progettare con

L'ELETTRONICA

ELETTRONICA INTEGRATA DIGITALE CON ESPERIMENTI

dai contatori alle memorie

GRUPPO EDITORIALE JACKSON

progettare con

L' ELETTRONICA

ELETTRONICA INTEGRATA DIGITALE CON ESPERIMENTI

dai contatori alle memorie

GRUPPO EDITORIALE
JACKSON

Direttore responsabile :
Paolo Reina

Direttore di divisione :
Roberto Pancaldi

Traduttore e impaginatore :
Eugenio Piana

Coordinamento editoriale :
Renata Rossi

Copertina :
Sergio Mazzali

Stampa :
GRAFICA 85 - Rodano Millepini

Distribuzione :
Sodip - Milano

Tutti i diritti di riproduzione e pubblicazione di disegni, fotografie e testi sono riservati.

© copyright per l'edizione originale
Ediciones Ingelek S.A.
© copyright per l'edizione italiana
Gruppo Editoriale Jackson - 1989

INDICE GENERALE

CAPITOLO	**1**	**Introduzione**	1
	1.1	Obbiettivi	1
CAPITOLO	**2**	**Contatori**	3
	2.1	Contatori asincroni	3
	2.2	Contatori divisori per N	5
	2.3	Contatori sincroni	6
	2.4	Contatori up-down	12
CAPITOLO	**3**	**Codici di numerazione**	19
	3.1	Codice binario	19
	3.2	Codice BCD	20
	3.3	Codice Gray	20
CAPITOLO	**4**	**Esercizi pratici sui contatori**	25
CAPITOLO	**5**	**Registri a scorrimento**	31
	5.1	Bistabile di tipo D	31
	5.2	Funzionamento dei registri a scorrimento	32
	5.3	Entrate ed uscite parallelo/serie	33
	5.4	Contatori ad anello	37
	5.5	Contatore Johnson	37
CAPITOLO	**6**	**Esercizi pratici sui registri a scorrimento**	39
CAPITOLO	**7**	**La piastra sperimentale**	43
	7.1	Fonte di alimentazione	43

	7.2	Componenti principali del circuito	46
	7.3	Consigli per l'uso	53
	7.4	Elenco dei componenti del preparatore digitale II	60
CAPITOLO	**8**	**Contatori integrati**	63
	8.1	Contatore decimale	63
	8.2	Codici utilizzati per la visualizzazione	65
	8.3	Decodificatori	66
	8.4	Decodificatore da BCD a 7 segmenti	71
CAPITOLO	**9**	**Unità aritmetico logica**	77
	9.1	Circuito integrato 4581	77
CAPITOLO	**10**	**Memoria RAM**	87
	10.1	Memorie RAM	87
	10.2	Memoria RAM statica 2114	90
CAPITOLO	**11**	**Codici alfanumerici e loro rappresentazione**	95
	11.1	Codice ASCII	95
	11.2	Display	97

CAPITOLO 1

INTRODUZIONE

Questo libro è la continuazione del primo volume, nel quale, a causa del poco spazio, non abbiamo completato lo studio dell'elettronica digitale. Questo non significa che senza aver letto il primo volume non si possa comprendere questo secondo, giacchè, per capire quanto contenuto in questo libro è sufficiente la sola conoscenza dei principi fondamentali.

1.1 Obbiettivi

Certamente non pretendiamo con questo secondo volume di esaurire l'argomento, ci proponiamo però di approfondire alcuni aspetti e di dare un'idea generale di alcuni argomenti che non sono stati trattati nel volume precedente, come: contatori, operatori aritmetici e memorie. Con questo libro intendiamo raggiungere questo obbiettivo.

In primo luogo tratteremo i contatori composti da elementi discreti, cioè i bistabili, che sono la base degli integrati. Parleremo di contatori binari classici, contatori ad anello, Johnson, registri a scorrimento, contatori sincroni e anche divisori di frequenza per numeri diversi dalla potenza di 2.

Per la realizzazione delle esercitazioni pratiche, che ci faranno capire perfettamente la teoria di questi argomenti, si disporrà della prima piastra per eser-

citazioni digitali e, dato che, non sfrutteremo appieno le possibilità di questa apparecchiatura, sarà possibile per l'appassionato andare oltre ed imparare molto di più sfruttando semplicemente la propria voglia di fare e la propria fantasia.

Dopo aver esposto questi temi, incominceremo ad usare la seconda piastra per esercitazioni digitali, che è sicuramente più complessa. Inizieremo con i contatori integrati decimali, coordinati con i decodificatori per visualizzatori a 7 segmenti. Esporremo anche i diversi codici di numerazione maggiormente utilizzati, ovviamente ne esistono molti di più e altri ancora possono essere inventati in funzione delle singole necessità di ogni applicazione.

Per capire il funzionamento di un calcolatore, vedremo come funziona una ALU (Unità Aritmetico-Logica), dato che, come circuito integrato o come elemento appartenente ad una CPU, è assolutamente necessaria per fare operazioni logiche e aritmetiche ad alta velocità.

Un altro elemento indispensabile in qualsiasi calcolatore è la memoria, per questa ragione cercheremo di far capire al lettore come funzionano le linee di entrata-uscita dati, di lettura-scrittura, di indirizzamento e selezione. Per far sì che possiate avere un'idea chiara del funzionamento dei calcolatori, verso i quali si orienta tutto il mondo dell'elettronica digitale, vi invitiamo a leggere attentamente le pagine di questo libro per poter così scoprire il fascino del funzionamento interno di tutti i componenti del calcolatore. Siamo spiacenti se a volte le spiegazioni risulteranno complesse, ma purtroppo, anche se la nostra intenzione è ovviamente di essere chiari e concisi, ciò non è sempre possibile.

CAPITOLO 2

CONTATORI

Abbiamo già visto nel primo volume "elettronica digitale con esperimenti", che i multivibratori (conosciuti nel mondo dell'elettronica digitale come "flip-flop", parola di origine inglese) possono dividere una frequenza qualsiasi per un multiplo di due, cioè, ottenere frequenze di periodo doppio, quadruplo, ecc. Questa proprietà è la caratteristica fondamentale dei contatori digitali. L'obbiettivo, è ottenere che, introdotti n impulsi in ingresso, le uscite dei flip-flop del contatore rappresentino, in binario o in un altro codice qualsiasi, il numero n di impulsi introdotti.

2.1 Contatori asincroni

Il contatore asincrono è il più semplice e per tanto il più facile da progettare. Si chiama asincrono perché non tutti i flip-flop che compongono il contatore sono controllati dallo stesso impulso di clock. L'uscita di ogni flip-flop si utilizza come ingresso di clock per il successivo.

In Fig. 2.1 è riportato un contatore asincrono modulo sedici formato da quattro flip-flop di tipo J-K, attivi sul fronte di discesa. Come si può vedere, ogni bistabile è collegato individualmente, in modo da dividere per 2 la frequenza del proprio impulso di clock. Così, il flip-flop A cambia di stato ad ogni im-

pulso di clock e per tanto divide la frequenza dello stesso per 2. Il flip-flop B cambia stato ad ogni impulso del proprio clock, dividendo per 2 la frequenza degli impulsi di Q_A, quindi la frequenza dell'uscita Q_B risulta divisa per quattro. Q_C risulta dunque divisa per 8 e Q_D per 16. Per tanto questi contatori dividono la frequenza dell'impulso di clock per 2^n, essendo n il numero di flip-flop che compongono il contatore.

In questo modo, si può dividere un qualunque multiplo di 2 mettendo il numero di flip-flop necessari. Questi circuiti si chiamano contatori perchè riportano in uscita il numero di impulsi che si introducono in ingresso. In effetti, mettendo in ordine le uscite Q_D, Q_C, Q_B e Q_A, di Fig. 2.1, si può verificare che

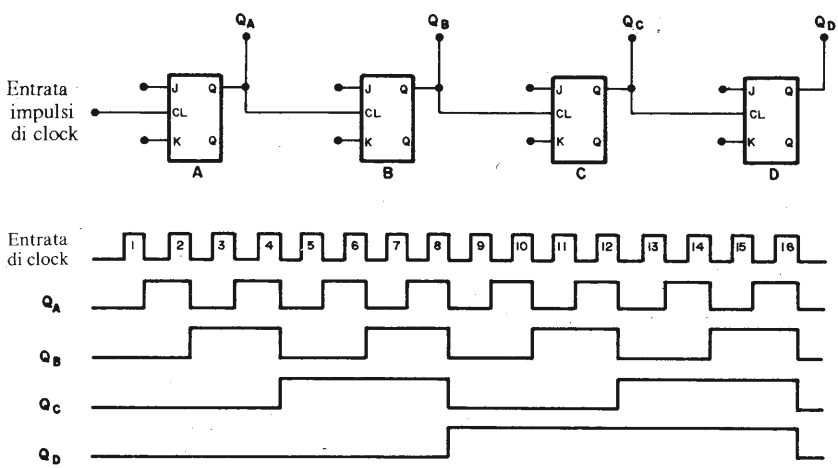


Fig. 2.1 *Contatore asincrono a quattro bit, formato da bistabili J-K attivi sul fronte di discesa.*

dopo l'arrivo del primo impulso di clock, gli stati di queste uscite sono 0001, che è il numero 1 in binario; dopo il secondo è 0010, che è il numero 2; dopo il terzo è 0011, che è il numero 3 e così di seguito. Concludiamo dicendo, che nelle uscite si può leggere in ogni momento il numero di impulsi che sono stati introdotti nel contatore.

2.2 Contatori divisori per N

Per fare una divisione, attraverso contatori asincroni, per un qualsiasi numero intero N che non sia multiplo di 2, bisogna usare i "PRESET" o i "CLEAR" dei flip-flop.

Le regole per ottenere il divisore usando i "PRESET" sono:

1. Trovare il numero n di flip-flop che si richiedono:

$$2^{n-1} < N \leq 2^n$$

2. Unire l'uscita di ogni flip-flop con l'entrata di clock del successivo.
3. Trovare il numero binario N-1.
4. Collegare tutte le uscite dei flip-flop che, nel conto N-1, hanno valore logico 1 ad una porta NAND. Portare alla porta NAND anche l'impulso di clock collegato in ingresso.
5. Collegare l'uscita della porta NAND ai "$\overline{\text{PRESET}}$" di tutti i flip-flop la cui uscita è 0 nel conto N-1. Se il "PRESET" dei flip-flop è attivo alto, come è il caso di quelli che contiene la nostra piastra digitale, logicamente la porta utilizzata sarà una AND.

In effetti, un contatore di n flip-flop quando arriva al suo massimo valore 2^n, ritorna allo stato iniziale. Di conseguenza quando il nostro contatore giunge nello stato N-1 i "PRESET" vengono attivati ed il contatore si porta nello stato 2^n, di conseguenza l'impulso N, li fa tornare allo stato iniziale. Per renderlo più chiaro facciamo un esempio di come si realizza un contatore modulo 10 seguendo questo metodo:

1. $2^3 < 10 < 2^4 \Rightarrow$ per tanto servono 4 flip-flop.
2. $N = 10 \Rightarrow$ in binario 1010.
3. $N-1 = 9 \Rightarrow$ in binario 1001.

I collegamenti, per tanto, saranno quelli indicati in Fig. 2.2. In questo modo in Q_D si ottiene una divisione per 10 della frequenza di clock presente in ingresso.

Le regole per ottenere il divisore usando i "RESET" o "CLEAR" dei flip-flop sono:

1. Trovare il numero di flip-flop richiesti:

$$2^{n-1} < N < 2^n$$

2. Unire l'uscita di ogni flip-flop con l'entrata di clock del successivo.
3. Trovare il numero binario N.
4. Collegare tutte le uscite dei flip-flop che sono a 1 nel numero N, agli ingressi di una porta NAND.
5. Collegare l'uscita della porta NAND a tutti i "$\overline{\text{CLEAR}}$" dei flip-flop. Se il "CLEAR" dei flip-flop è attivo alto, come è il caso di quelli che contiene la nostra piastra digitale, logicamente la porta utilizzata sarà una AND.

In effetti, in questo modo quando il contatore arriva allo stato N, si azzera.

Per maggiore chiarezza andiamo a vedere un esempio di come si realizza un contatore modulo 6 seguendo questo metodo:

1. $2^2 < 6 < 2^3$ per tanto servono 3 flip-flop.
2. N = 6 in binario 110

I collegamenti saranno quelli indicati in Fig. 2.3. In Q_C si ottiene la frequenza di clock presente in ingresso divisa per 6.

2.3 Contatori sincroni

Quando il numero di flip-flop che compongono un contatore è grande, il contatore asincrono può generare errori dovuti all'accumulo dei tempi di ritardo di tutti i suoi flip-flop.

I contatori sincroni sono caratterizzati dal fatto che tutti i flip-flop sono controllati dallo stesso impulso di clock e così si elimina il possibile errore precedentemente citato. La progettazione di questi contatori non è così semplice come nel caso degli asincroni, ma si semplifica usando le mappe di Karnaugh.

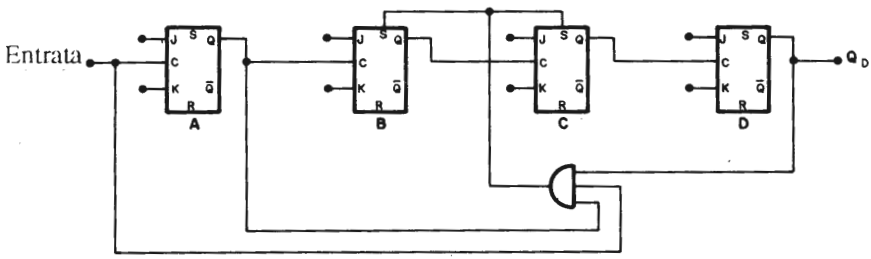


Fig. 2.2 *Divisore per 10 ottenuto con il metodo che utilizza i set o preset.*

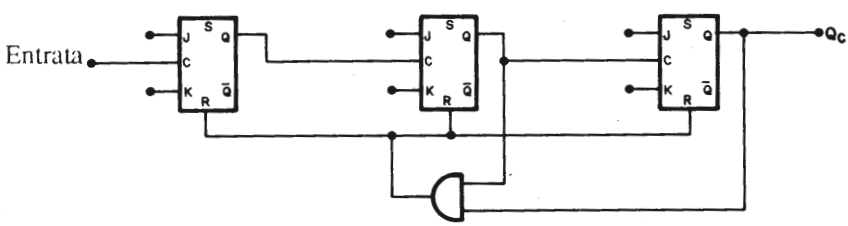


Fig. 2.3 *Divisore per 6 ottenuto con il metodo che utilizza i reset.*

La Fig. 2.4 mostra un contatore sincrono formato da 4 flip-flop, come si può vedere, l'ingresso di clock è comune a tutti i flip-flop.

In funzione della numerazione binaria, il flip-flop A deve cambiare stato ad ogni impulso di clock, per questo le sue entrate J e K sono a 1. Il flip-flop B deve cambiare stato quando $Q_A = 1$. Il C cambia stato quando $Q_A = Q_B = 1$ e il D quando $Q_A = Q_B = Q_C = 1$.

Per realizzare il progetto di un divisore sincrono per un numero qualsiasi si utilizzano le mappe di Karnaugh. Andiamo a vedere, per esempio, il progetto di un divisore per 10 o contatore modulo 10 formato da flip-flop di tipo J-K.

In primo luogo osserviamo la tabella di eccitazione del flip-flop J-K.

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

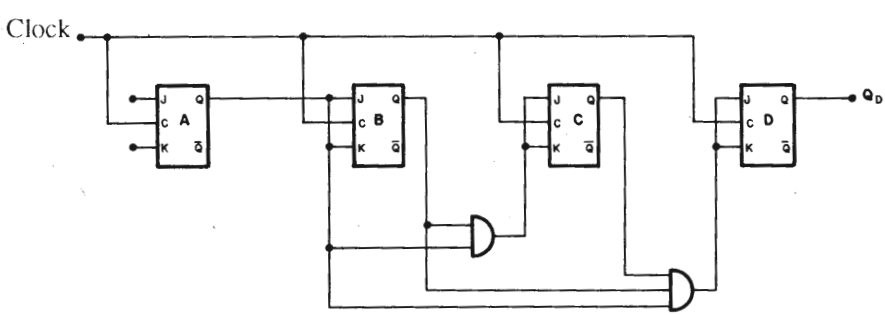


Fig. 2.4 *Contatore sincrono a 4 bit.*

Questa si ricava dalla tabella di verità corrispondente. Il significato è il seguente: se abbiamo in Q uno 0 e vogliamo che dopo un impulso di clock continui ad avere uno 0, l'entrata J deve essere a 0, mentre l'entrata K non importa in quale stato si trova. Infatti secondo la tabella di verità, per J = 0 e K = 0 si otterrà $Q_n = Q_{n+1}$ e per J = 0 e K = 1 risulterà $Q_{n+1} = 0$. Allo stesso modo, se si ha in Q uno 0 e si vuole un 1 in seguito ad un nuovo impulso di clock, J deve avere valore 1 mentre il valore di K è indifferente; se si desidera far commutare Q da 1 a 0, il valore di J è indifferente mentre quello di K dovrà essere 1; se si vuole mantenere la Q a 1, il valore di J è indifferente mentre K dovrà essere 0.

Passiamo ora al diagramma degli stati del divisore per 10:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	X	X	X	X
10	8	9	X	X

Per realizzare un divisore per 10 sono necessari 4 flip-flop, dato che:

$$2^3 < 10 < 2^4$$

Bisognerà fare 4 mappe di Karnaugh, una per il controllo di ogni flip-flop. Queste mappe sono le seguenti:

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	1X	X1	X1	1X
01	1X	X1	X1	1X
11	XX	XX	XX	XX
10	1X	X1	XX	XX

Flip-flop A

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0X	1X	X1	X0
01	0X	1X	X1	X0
11	XX	XX	XX	XX
10	0X	0X	XX	XX

J_BK_B

Flip-flop B

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0X	0X	1X	0X
01	X0	X0	X1	X0
11	XX	XX	XX	XX
10	0X	0X	XX	XX

J_CK_C

Flip-flop C

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0X	0X	0X	0X
01	0X	0X	1X	0X
11	XX	XX	XX	XX
10	X0	X1	XX	XX

J_DK_D

Flip-flop D

Tali mappe si realizzano nel seguente modo: supponiamo di trovarci nello stato 2, in modo tale che lo stato seguente dovrà essere il 3. Questi stati sono:

- Stato 2: $Q_DQ_CQ_BQ_A = 0010$
- Stato 3: $Q_DQ_CQ_BQ_A = 0011$

Il flip-flop A deve passare da 0 a 1, il che accadrà, come si può notare nella tabella di eccitazione, quando J vale 1, in modo che nella casella dello stato 2 del flip-flop, metteremo X0.

Allo stesso modo, in quelle dei flip-flop C e D si metterà 0X. Così, passando da stato a stato si vanno completando i diagrammi, in modo da ottenere:

- Per il flip-flop A: $J_A = K_A = 1$
- Per il flip-flop B: $J_B = K_B = Q_A \cdot \overline{Q}_D$
- Per il flip-flop C: $J_C = K_C = Q_A \cdot Q_B$
- Per il flip-flop D: $J_D = Q_A \cdot Q_B \cdot Q_C = J_C \cdot Q_C$

  $K_D = Q_A$

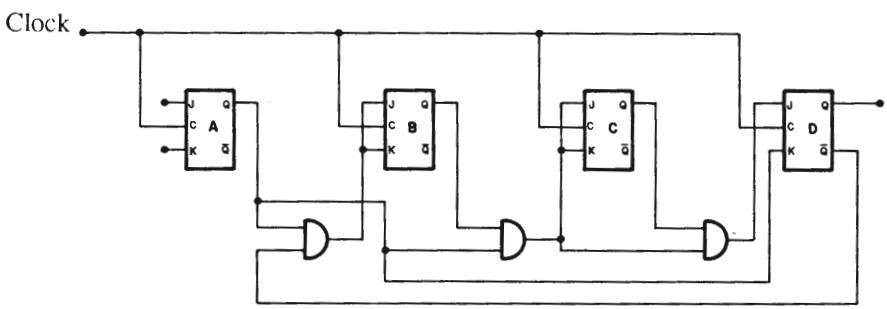


Fig. 2.5 *Contatore decimale sincrono.*

Come si può vedere, gli stati nei quali il contatore non deve commutare, si completano con XX. In funzione delle equazioni ottenute, il contatore sincrono modulo 10 dovrà essere realizzato come mostrato in Fig. 2.5.

2.4 Contatori up-down

In molte applicazioni, il contatore sincrono deve essere capace di contare in senso ascendente o discendente. Questi contatori sono chiamati "*up-down*", termine inglese che significa su e giù. Per avere queste caratteristiche bisogna introdurre una nuova variabile, che chiameremo Z, con la quale si seleziona il senso del conteggio, in modo che, per esempio, se Z = 1 si conta in modo ascendente, se Z = 0 si conta in modo discendente.

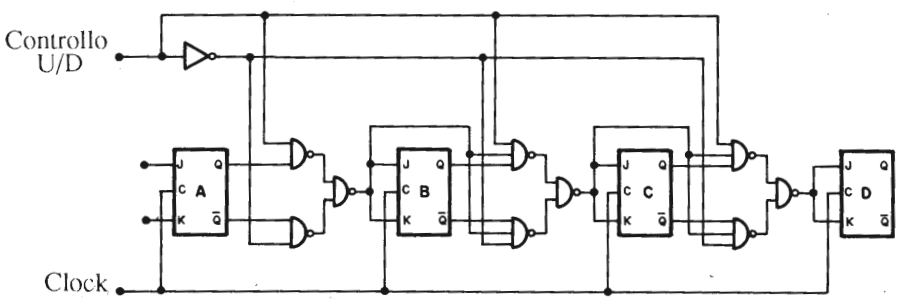


Fig. 2.6 *Contatore binario sincrono a quattro bit ascendente/discendente (up-down).*

Per progettare questi contatori bisogna realizzare le mappe di Karnaugh per ogni flip-flop una per Z = 1 e un altra per Z = 0. Queste mappe saranno diverse, dato che il diagramma degli stati per Z = 1 avrà un senso di conteggio e per Z = 0 il contrario.

Il risultato finale, a partire dai corrispondenti diagrammi, sarà:

$$J_i = Z \cdot f_{1i}\,(Q_A,Q_B,Q_C,Q_D) + \overline{Z} \cdot f_{2i}\,(Q_A,Q_B,Q_C,Q_D)$$

$$K_i = Z \cdot f_{3i}\,(Q_A,Q_B,Q_C,Q_D) + \overline{Z} \cdot f_{4i}\,(Q_A,Q_B,Q_C,Q_D)$$

essendo: f_{1i} la funzione risultante per J_i nel diagramma di Z = 1,
f_{2i} la funzione risultante per J_i nel diagramma di Z = 0,
f_{3i} la funzione risultante per K_i nel diagramma di Z = 1,
f_{4i} la funzione risultante per K_i nel diagramma di Z = 0.

Vediamo come esempio il progetto di un contatore binario sincrono "up-down" formato da 4 flip-flop. I corrispondenti diagrammi degli stati sono:

Z = 1

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

Z = 0

Q_DQ_C \ Q_BQ_A	00	01	11	10
00	0	15	13	14
01	12	11	9	10
11	4	3	1	2
10	8	7	5	6

Utilizzando flip-flop di tipo JK, in accordo con la loro tabella di eccitazione e con i diagrammi degli stati, si ottengono le mappe di ogni flip-flop.

Per il flip-flop A:

Q_DQ_C \ Q_BQ_A (Z = 1)	00	01	11	10
00	1X	X1	X1	1X
01	1X	X1	X1	1X
11	1X	X1	X1	1X
10	1X	X1	X1	1X

Q_DQ_C \ Q_BQ_A (Z = 0)	00	01	11	10
00	1X	X1	X1	1X
01	1X	X1	X1	1X
11	1X	X1	X1	1X
10	1X	X1	X1	1X

Per tanto: $J_A = K_A = 1$

Per il flip-flop B:

Q_DQ_C \ Q_BQ_A (Z = 1)	00	01	11	10
00	0X	1X	X1	X0
01	0X	1X	X1	X0
11	0X	1X	X1	X0
10	0X	1X	X1	X0

Q_DQ_C \ Q_BQ_A (Z = 0)	00	01	11	10
00	1X	0X	X0	X1
01	1X	0X	X0	X1
11	1X	0X	X0	X1
10	1X	0X	X0	X1

Per tanto: $J_B = K_B = Z \cdot Q_A + \overline{Z} \cdot \overline{Q}_A$

Per il flip-flop C:

Q_DQ_C \ Q_BQ_A (Z = 1)	00	01	11	10
00	0X	0X	1X	0X
01	X0	X0	X1	X0
11	X0	X0	X1	X0
10	0X	0X	1X	0X

Q_DQ_C \ Q_BQ_A (Z = 0)	00	01	11	10
00	1X	0X	0X	0X
01	X1	X0	X0	X0
11	X1	X0	X0	X0
10	1X	0X	0X	0X

Per tanto $J_C = K_C = Z \cdot Q_A \cdot Q_B + \overline{Z} \cdot \overline{Q}_A \cdot \overline{Q}_B$

Per il flip-flop D:

Q_DQ_C \ Q_BQ_A (Z = 1)	00	01	11	10
00	0X	0X	0X	0X
01	0X	0X	1X	0X
11	X0	X0	X1	X0
10	X0	X0	X0	X0

Q_DQ_C \ Q_BQ_A (Z = 0)	00	01	11	10
00	1X	0X	0X	0X
01	0X	0X	0X	0X
11	X0	X0	X0	X0
10	X1	X0	X0	X0

Per tanto: $J_D = K_D = Z \cdot Q_A \cdot Q_B \cdot Q_C + \overline{Z} \cdot \overline{Q}_A \cdot \overline{Q}_B \cdot \overline{Q}_C$

Con queste equazioni si può già ottenere lo schema del contatore riportato in Fig. 2.6.

AMPLIAMENTO DEI CONTATORI DECIMALI

I contatori binari possono essere aumentati semplicemente aggiungendo flip-flop. L'unica cosa da farsi per ottenere un contatore che conti unità, decine, centinaia, ecc., in BCD (decimale codificato in binario), è disporre di vari contatori modulo 10, come quelli già spiegati, in modo tale che l'uscita D comandi la decade successiva, potendosi così costruire contatori BCD di qualsiasi lunghezza.

CONTATORI PRESELEZIONABILI

In alcune occasioni è necessario utilizzare contatori che non inizino il conteggio da zero, ma da un valore predeterminato. In questi casi, quando il contatore raggiunge il numero massimo del conteggio non si deve applicare uno zero logico alle entrate clear (reset o inizio) di tutti i flip-flop, ma bensì applicare stati logici 0 alle entrate di reset di quei flip-flop che devono restare a 0 e alle entrate preset (entrata set diretta) di quelli che devono restare a 1 per realizzare il numero di conteggio iniziale. Si può anche procedere in modo diverso, prima si azzerano tutte le uscite del contatore e in un secondo tempo si mettono a 1 solo alcune uscite, in funzione del valore del numero iniziale.

CAPITOLO 3

CODICI DI NUMERAZIONE

Quando per la realizzazione di un progetto abbiamo bisogno di contatori, la prima cosa che dobbiamo fissare è il codice di numerazione che adopereranno gli stessi. Esistono diversi codici di numerazione, alcuni li abbiamo già descritti nel precedente libro che trattava questo tema, li ripasseremo tra breve; inoltre ne esporremo di nuovi usati di frequente.

La scelta del codice è operata in base alla sua comodità d'uso. Di seguito esponiamo i più usati.

3.1 Codice binario

Il codice binario è il sistema binario di numerazione.

0 - 0000	9 - 1001
1 - 0001	10 - 1010
2 - 0010	11 - 1011
3 - 0011	12 - 1100
4 - 0100	13 - 1101
5 - 0101	14 - 1110
6 - 0110	15 - 1111
7 - 0111	16 - 10000
8 - 1000	

3.2 Codice BCD

Il codice BCD (binary coded decimal, che significa decimale codificato in binario) è una specie di miscela tra il sistema decimale e il binario. Ad ogni digit del sistema decimale viene assegnato un codice di quattro bit del sistema binario. In tal modo, fino al numero 9, il codice BCD è uguale al binario, a partire dal 10 cambia.

0 - 0000
1 - 0001
2 - 0010
3 - 0011
4 - 0100
5 - 0101
6 - 0110
7 - 0111
8 - 1000
9 - 1001
10 - 10000
11 - 10001
12 - 10010
13 - 10011

Per esempio, il numero 326 del nostro sistema decimale abituale, in codice BCD sarà:

0011	0010	0110
3	2	6

In questo modo, il numero di bit necessari per rappresentare un numero in codice BCD è superiore a quello del binario. Tuttavia questo codice è molto usato poichè è più facilmente interpretabile rispetto al sistema binario puro.

3.3 Codice Gray

E' un codice continuo e ciclico, in modo che le configurazioni corrispondenti a numeri decimali consecutivi sono adiacenti, essendo anche adiacente l'ul-

tima combinazione con la prima. Il codice Gray è caratterizzato dal fatto che due numeri consecutivi si differenziano in un solo bit.

Per quattro bit questo codice è:

0 - 0000
1 - 0001
2 - 0011
3 - 0010
4 - 0110
5 - 0111
6 - 0101
7 - 0100
8 - 1100
9 - 1101
10 - 1111
11 - 1110
12 - 1010
13 - 1011
14 - 1001
15 - 1000

Il principale utilizzo di questo codice è nei convertitori analogico-digitali e nei codificatori di posizione che convertono un angolo in una combinazione binaria attraverso un disco codificato che in inglese viene chiamato "shaft encoders". In base alla sua principale proprietà, che è: una combinazione binaria e la contigua cambia in un solo bit, viene eliminata la possibilità di apparizione di combinazioni intermedie che possono provocare un'informazione erronea nel momento in cui si realizza una misura.

Il codice Gray si forma a partire dal binario come di seguito si espone:

$$B_nB_{n-1}....B_3B_2B_1B_0 \qquad G_nG_{n-1}....G_3G_2G_1G_0$$

dove B è in binario e G in Gray, entrambi rappresentano lo stesso numero decimale N, e:

$$G_0 = B_0 \oplus B_1$$
$$G_1 = B_1 \oplus B_2$$
$$G_2 = B_2 \oplus B_3$$

in generale:

$$G_n = B_n \oplus B_{n+1}$$

dove $\oplus$ è la funzione "OR esclusiva".

CODICI AUTOCOMPLEMENTARI

Un codice è autocomplementare quando la combinazione corrispondente al complemento a 9 di N, cioè 9–N, è ottenuto invertendo la combinazione corrispondente a N in binario, cioè, cambiando gli uno con zeri e viceversa.

I codici di questo tipo più usati sono due: codice Eccesso-3 e 2-4-2-1.

Eccesso 3	2-4-2-1
0 - 0011 (–9)	0 - 0000 (–9)
1 - 0100 (–8)	1 - 0001 (–8)
2 - 0101 (–7)	2 - 0010 (–7)
3 - 0110 (–6)	3 - 0011 (–6)
4 - 0111 (–5)	4 - 0100 (–5)
5 - 1000 (–4)	5 - 1011 (–4)
6 - 1001 (–3)	6 - 1100 (–3)
7 - 1010 (–2)	7 - 1101 (–2)
8 - 1011 (–1)	8 - 1110 (–1)
9 - 1100 (–0)	9 - 1111 (–0)

CODICI PER LA CORREZIONE DI ERRORI

Nella trasmissione di un'informazione numerica, è possibile che si verifichino errori dovuti a rumori o a guasti nei componenti per questa ragione si utilizzano codici speciali con i quali è possibile rilevare questo tipo di errore attraverso un semplice controllo. Uno di questi codici è chiamato 2 su 5, nel quale ogni numero è una combinazione di 5 bit dei quali due sono a valore logico 1. Quando si opera il controllo e si trova un numero con una quantità diversa di numeri uno, significa che è stato introdotto un errore.

Il codice 2 su 5 è il seguente:

0 - 11000
1 - 00011
2 - 00101
3 - 00110
4 - 01001
5 - 01010
6 - 01100
7 - 10001
8 - 10010
9 - 10100

CAPITOLO 4

ESERCIZI PRATICI SUI CONTATORI

Con l'intento di assimilare bene le idee esposte nei precedenti capitoli sui contatori, realizzeremo ora quattro esercitazioni che serviranno a chiarirci alcuni concetti. Data la semplicità delle stesse si realizzeranno sulla prima piastra digitale.

La prima esercitazione, anche se non è propriamente sui contatori, è comunque necessaria, dato che realizzeremo un generatore di clock indispensabile per svolgere le successive esercitazioni.

ESERCITAZIONE 1. IL GENERATORE DI CLOCK

1. Realizzate il circuito generatore di clock di Fig. 4.1. Contate il numero di volte che si accende il diodo LED in 30 secondi e ricavate la frequenza del generatore di impulsi attraverso la formula:

$$f = \frac{\text{num. impulsi contati}}{30} = N\ \text{Hz}$$

f deve risultare approssimativamente 1 Hz, può variare a causa della tolleranza dei componenti.

ESERCITAZIONE 2. CONTATORE ASINCRONO DI 4 BIT

I. Montate il contatore di Fig. 4.2.

II. Quando applicate la tensione di alimentazione, il contatore potrà trovarsi in un qualsiasi stato, che dipenderà dalle caratteristiche interne dei bistabili utilizzati. Questo stato arbitrario deve essere cancellato, ponendo momentaneamente a zero gli ingressi di inizializzazione (clear).

III. Collegate le porte NAND 1 e 2 come flip-flop RS e utilizzatelo per inviare impulsi all'ingresso del primo bistabile, che è anche l'ingresso del contatore. Otterrete la tabella di verità che segue:

FF1	FF2	FF3	FF4	Impulso di clock
0	0	0	0	inizializzazione
1	0	0	0	1.º impulso
0	1	0	0	2.º impulso
1	1	0	0	3.º impulso
0	0	1	0	4.º impulso
1	0	1	0	5.º impulso
0	1	1	0	6.º impulso
1	1	1	0	7.º impulso
0	0	0	1	8.º impulso
1	0	0	1	9.º impulso
0	1	0	1	10.º impulso
1	1	0	1	11.º impulso
0	0	1	1	12.º impulso
1	0	1	1	13.º impulso
0	1	1	1	14.º impulso
1	1	1	1	15.º impulso
0	0	0	0	16.º impulso

IV. Conclusione: secondo questa tabella, un contatore asincrono di 4 bit ha 16 diverse configurazioni e, di conseguenza, conta in sequenza binaria da 0000 fino a 1111.

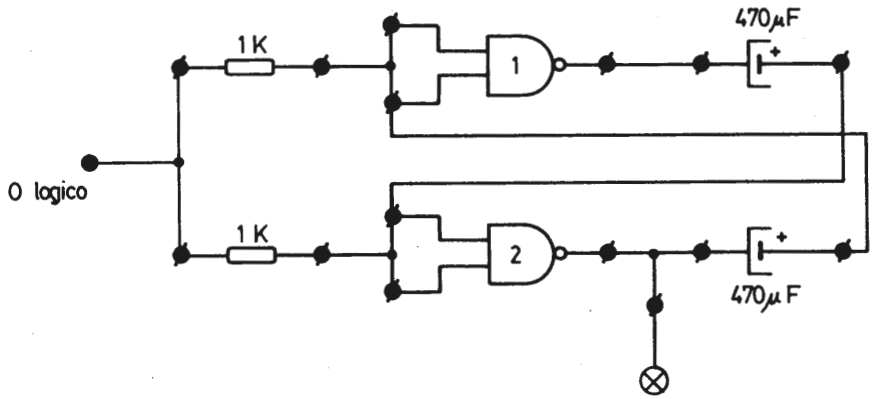


Fig. 4.1 *Schema di collegamento dell' esercitazione 1.*

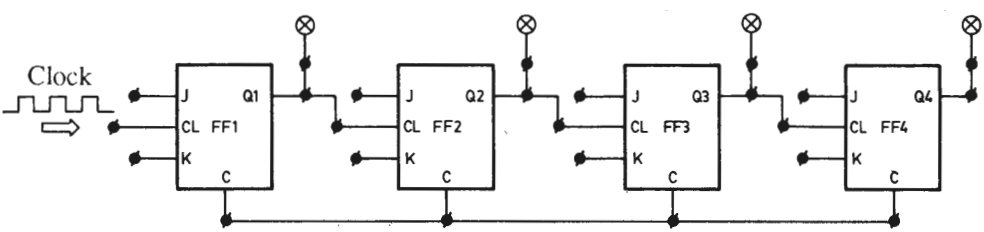


Fig. 4.2 *Schema di collegamento dell' esercitazione 2.*

ESERCITAZIONE 3. DIVISORE ASINCRONO PER 6

I. Montate il circuito di Fig. 4.3.

II. Applicando la tensione di alimentazione si verifica la stessa cosa accaduta nell'esercitazione precedente.

III. Collegate le porte NAND 1 e 2 come nell'esercitazione precedente. Allo stesso modo, inviando impulsi all'ingresso di clock si otterrà la tabella di verità che segue:

FF1	FF2	FF3	FF4	Impulso di clock
0	0	0	0	inizializzazione
1	0	0	0	1.º impulso
0	1	0	0	2.º impulso
1	1	0	0	3.º impulso
0	0	1	0	4.º impulso
1	0	1	0	5.º impulso

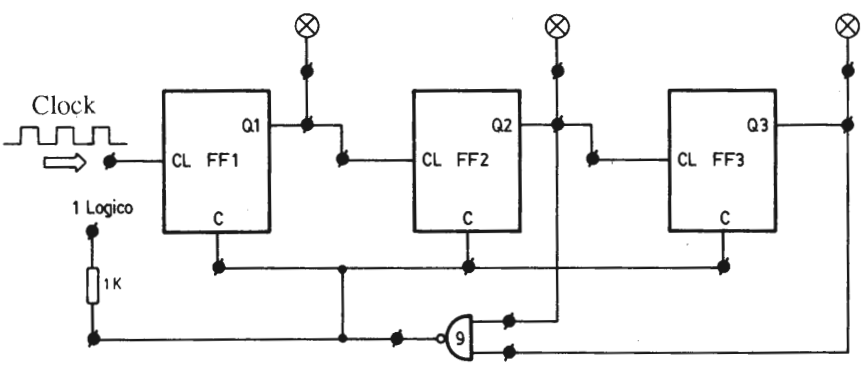


Fig. 4.3 *Schema di collegamento dell' esercitazione 3.*

IV. Concludendo possiamo notare che la frequenza di clock che inviamo all'ingresso rimane divisa in Q_3 per 6. Applicando l'impulso di clock della prima esercitazione otterremo una frequenza approssimata di 1/6 Hz. Questo contatore conta in sequenza binaria da 000 fino a 101.

ESERCITAZIONE 4. CONTATORE SINCRONO DI 2 BIT

I. Montate il circuito di Fig. 4.4.

II. Inizializzate il sistema.

III. Collegate il generatore di clock dell'esercitazione 1 all'ingresso. Si osserverà che il LED A si accende ogni 2 secondi e il B ogni 4.

IV. Conclusione: il funzionamento è identico a quello di un contatore asincrono, ma ci evita problemi di ritardo nella logica che controlla i cambiamenti di stato quando le frequenze sono elevate.

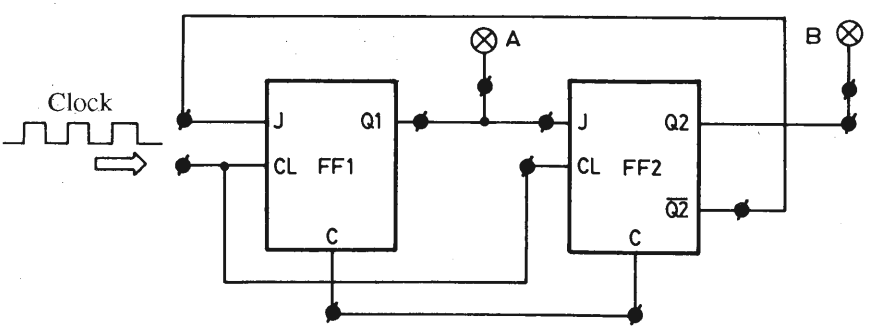


Fig. 4.4 *Schema di collegamento dell'esercitazione 4.*

V. Se per terminare introduciamo gli impulsi uno ad uno, come abbiamo fatto nell'esercitazione precedente, otterremo la seguente tabella di verità.

A	**B**	**Impulso di clock nº**
0	0	0
1	0	1
0	1	2
0	0	3

CAPITOLO 5

REGISTRI A SCORRIMENTO

I registri a scorrimento sono anche chiamati shift register, a causa della loro capacità di spostare o "shiftare" i dati al loro interno in avanti o indietro.

Tali registri sono costituiti da una catena di flip-flop opportunamente collegati tra loro.

Molto spesso sono utilizzati per la trasmissione dati.

5.1 Bistabile di tipo D

Questo bistabile o flip-flop è caratterizzato da un ingresso dati sincrono, chiamato D e da un ingresso di clock. Normalmente è provvisto anche di ingressi asincroni set e reset, come mostrato in Fig. 5.1.

Con l'arrivo di un impulso di clock l'uscita Q assume lo stesso stato che ha l'ingresso D in quel momento e lo mantiene fino all'arrivo di un nuovo impulso di clock che lo modifichi. Questo tipo di flip-flop si utilizza soprattutto quando si vogliono immagazzinare dei dati per periodi di tempo prefissati.

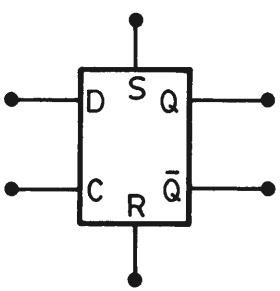
In Fig. 5.2 possiamo osservare come realizzare un flip-flop di tipo D a partire da un JK. Quest'ultimo tipo di flip-flop è già montato sulla nostra piastra digitale.

5.2 Funzionamento dei registri a scorrimento

Un registro a scorrimento è costituito da una catena di flip-flop collegati l'uno dietro l'altro.

L'uscita di ogni flip-flop è collegata all'ingresso del successivo; si applica un impulso di clock comune a tutti i flip-flop, in modo tale da ottenere un funzionamento sincrono.

In Fig. 5.3 è riportato un registro a scorrimento che utilizza bistabili di tipo D. Il dato che si introduce in ingresso shifta di un posto ad ogni impulso di clock, fino ad arrivare in uscita.



t_n	t_{n+1}
D	Q
0	0
1	1

Fig. 5.1 *Flip-flop o bistabile di tipo D.*

Supponiamo di avere un 1 in ingresso, in corrispondenza del primo impulso di clock il dato verrà trasferito sull'uscita Q del primo bistabile e sarà disponibile come ingresso D del secondo. Con il secondo impulso di clock l'uscita del secondo bistabile va a 1 e diventa l'ingresso del terzo. Con il successivo impulso il dato viene trasferito sull'uscita del terzo bistabile, diventando l'ingresso del quarto e così di seguito.

Il registro a scorrimento serve quindi per spostare l'informazione in modo seriale, oppure per introdurre un ritardo nell'informazione di tanti impulsi di clock quanti sono i bistabili che formano il registro.

5.3 Entrate ed uscite parallelo/serie

Il registro a scorrimento precedentemente analizzato (Fig. 5.3) è con ingresso ed uscita in serie. In un caso più generale, gli ingressi e le uscite possono essere sia di dati in serie che di dati in parallelo.

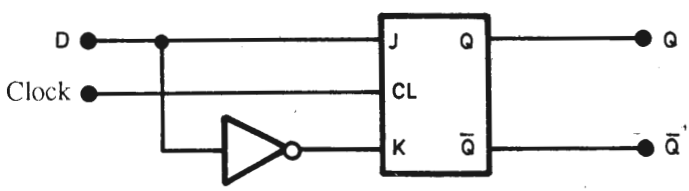


Fig. 5.2 *Conversione di un bistabile di tipo J-K in uno di tipo D.*

Consideriamo il registro a scorrimento di Fig. 5.4.
Questo registro ha diversi modi di funzionamento:

1. Ingresso serie-uscita serie. I dati entrano in serie dall'ingresso "serie" e ad ogni impulso di clock si spostano di un posto a destra, fino ad uscire di nuovo in serie dall'uscita Q_D. In questo modo si ottiene un ritardo di quattro impulsi di clock nell'informazione introdotta.

2. Ingresso parallelo-uscita parallela. I dati entrano in parallelo negli ingressi J_A, J_B, J_C e J_D. Inviando un segnale alto ai preset dei quattro flip-flop i dati vengono trasferiti alle uscite Q_A, Q_B, Q_C e Q_D. In questo modo il registro com-

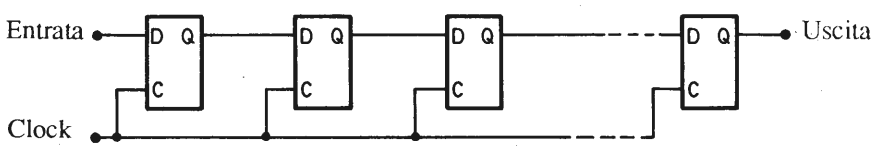


Fig. 5.3 *Registro a scorrimento realizzato con bistabili di tipo D.*

pie un immagazzinamento temporale di dati, svolgendo la funzione di latch.

3. Ingresso parallelo-uscita serie. I dati entrano in parallelo negli ingressi J_A, J_B, J_C e J_D. In modo analogo al precedente, inviando un segnale alto ai quattro preset i dati vengono trasferiti alle uscite Q_A, Q_B, Q_C e Q_D. A questo punto inviando degli impulsi di clock, i dati che sono stati introdotti in parallelo, cominciano ad uscire in serie dall'uscita Q_D. In questo modo, il registro a scorrimento compie una conversione di dati da parallelo a serie.

4. Ingresso serie-uscita parallela. I dati entrano in serie dall'ingresso "serie" inviando quattro impulsi di clock. I dati usciranno in formato parallelo dalle

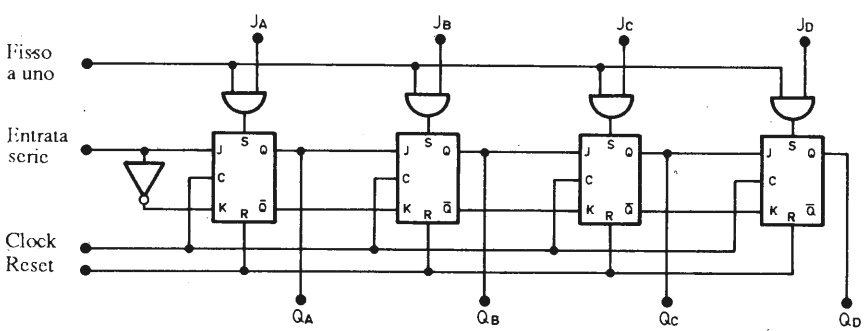


Fig. 5.4 *Registro a scorrimento.*

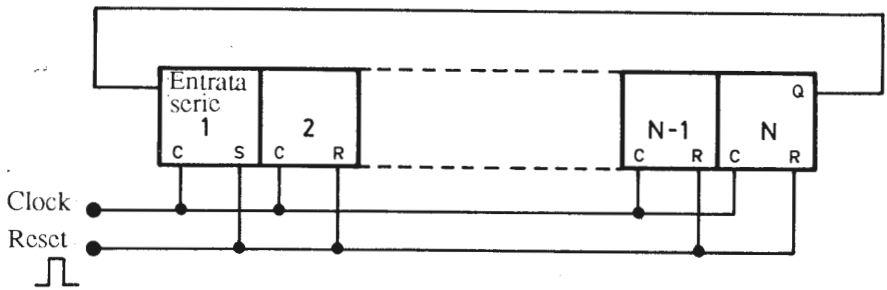


Fig. 5.5 *Contatore ad anello con spostamento di livelli alti.*

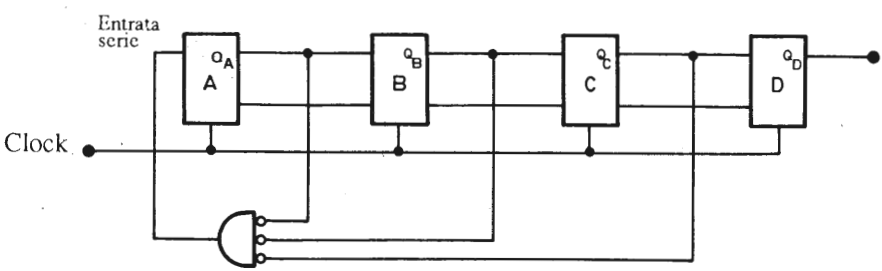


Fig. 5.6 *Sistema di rialimentazione più complesso che assicura il ritorno agli schemi validi con un massimo di quattro impulsi di clock.*

uscite Q_A, Q_B, Q_C e Q_D. Il registro a scorrimento compie dunque una conversione di dati da serie a parallelo.

5.4 Contatori ad anello

Il contatore ad anello è sostanzialmente formato da un registro a scorrimento, che presenta caratteristiche di funzionamento diverse rispetto ai registri analizzati in precedenza. Il contatore ad anello infatti opera lo spostamento di un solo livello logico 1 o 0, ottenendo così, uscite già decodificate senza seguire una logica di tipo addizionale.

Questo contatore si realizza collegando l'uscita dell'ultimo bistabile del registro a scorrimento all'ingresso del primo. In Fig. 5.5 è riportato un contatore ad anello con spostamento di livelli alti. Il contatore ad anello deve essere inizializzato ponendo a 1 il primo bistabile, a 0 i seguenti. In Fig. 5.6 è riportato un sistema di rialimentazione più complesso che garantisce il ritorno agli stati validi, con un massimo di quattro impulsi di clock. La logica di rialimentazione fa si che giunga un 1 al primo bistabile soltanto quando le uscite Q_A, Q_B e Q_C sono a zero. In questo modo non è necessaria l'operazione di inizializzazione, dato che il contatore si autocorregge dopo un po' di tempo.

Impulso di clock	**FLIP-FLOP Nº**				
	1	**2**	**....**	**N-1**	**N**
0	1	0	....	0	0
1	0	1	....	0	0
2	0	0	0	0	0
.	.	.		.	.
.	.	.		.	.
.	.	.		.	.
N-1	0	0	....	0	1
N	1	0	....	0	0

5.5 Contatore Johnson

Il contatore Johnson si differenzia dal contatore ad anello, poichè all'ingresso del bistabile viene collegato il complemento dell'ultimo flip-flop del registro. Il risultato è un contatore con 2N stati, come si può vedere in Fig. 5.7.

Questi stati sono quelli che formano il codice Johnson. Il quale per N stati assume le seguenti configurazioni:

	1	**2**	**3**	**....**	**N-1**	**N**
0	0	0	0	....	0	0
1	1	0	0	....	0	0
2	1	1	0	....	0	0
3	1	1	1	....	0	0
.						
.						
.						
.						
N–1	1	1	1	....	1	0
N	1	1	1	....	1	1
N+1	0	1	1	....	1	1
N+2	0	0	1	....	1	1
N+3	0	0	0	....	1	1
.						
.						
.						
.						
2N-1	0	0	0	....	0	1
2N	0	0	0	....	0	0

Il contatore Johnson deve essere inizializzato per ottenere la sequenza valida. Quest'inizializzazione consiste nell'inviare un impulso agli ingressi di reset di tutti i bistabili del registro.

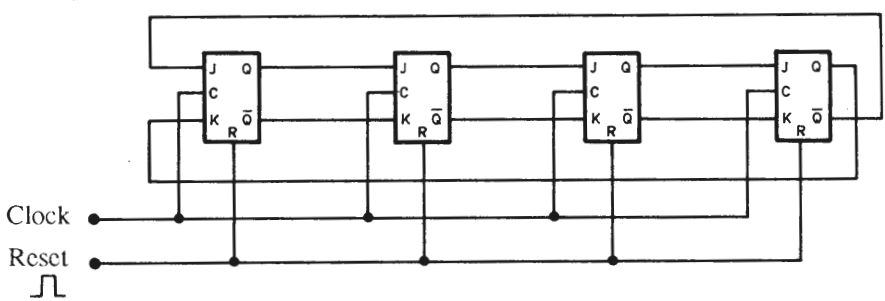


Fig. 5.7 *Contatore Johnson.*

CAPITOLO 6

ESERCIZI PRATICI SUI REGISTRI A SCORRIMENTO

In questo capitolo eseguiremo quattro esercitazioni sui registri a scorrimento, con l'intento di verificare praticamente il funzionamento degli stessi. Si eseguiranno sulla prima piastra digitale, questa contiene quattro bistabili di tipo JK che ci serviranno per costruire un registro a scorrimento di quattro bit. Vedremo anche un contatore ad anello e un contatore Johnson.

ESERCITAZIONE 1. REGISTRI A SCORRIMENTO

I. Realizzate il circuito di Fig. 6.1, composto dai quattro bistabili di tipo JK presenti sulla piastra digitale e da tre porte NAND. Anche se non è necessario, consigliamo di visualizzare gli stati delle uscite dei bistabili, attraverso i LED. Dopo aver applicato la tensione di alimentazione, devono essere inizializzati tutti i bistabili mettendo a zero le entrate (clear).

II. Inviare un 1 logico all'ingresso segnato con I e usare il bistabile RS, formato dalle porte NAND 1 e 2, per applicare un impulso di clock. Mettere a 0

logico l'ingresso I e applicare un altro impulso di clock. Mettere a 1 l'ingresso I e inviare due impulsi di clock. Dopo ogni impulso prendere nota delle uscite Q_1 e Q_4. Se tutto è stato eseguito correttamente otterrete la tabella di verità che segue:

Clock	A	B	C	D
inizio	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	0	1	0
4	1	1	0	1

III. Porre l'entrata a zero e azzerare tutti i bistabili applicando diversi impulsi di clock. Per avere in uscita 0000 saranno necessari minimo quattro impulsi.

IV. Porre l'entrata a 1 e compiere la stessa operazione eseguita precedentemente. Dovrà apparire 1111 in uscita dopo quattro impulsi di clock. Questi appariranno in sequenza, nel primo bistabile, poi nei primi due e così via fino all'accensione dei quattro LED.

ESERCITAZIONE 2. CONTATORE AD ANELLO

I. Per realizzare un contatore ad anello basta realizzare il circuito di Fig. 6.1, ricordandosi di collegare l'uscita Q_4 all'ingresso del circuito segnato con I. Prima di compiere qualsiasi operazione assicurarsi di aver inizializzato i bistabili. Applicare, come sempre, diversi impulsi di clock e osservare gli stati dell'uscita.

II. Se è stato eseguito il precedente passo avrete osservato che non succede niente, dato che un contatore ad anello non parte automaticamente. In realtà accade che circolano livelli zero in tutte le uscite del contatore.

III. Inviare per un istante, uno 0 logico all'ingresso (preset) del bistabile FF1. Applicare diversi impulsi di clock e osservare gli stati delle uscite. Otterrete la tabella di verità che segue:

Clock	A	B	C	D
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0

ESERCITAZIONE 3. REGISTRO AD ANELLO

I. Si esegua il montaggio come nel punto I dell'esercitazione precedente.

II. Di seguito collegare l'ingresso P (preset) del bistabile FF1 e C (clear) del bistabile FF4 ad un livello zero, per immagazzinare il numero 1001. Applica-

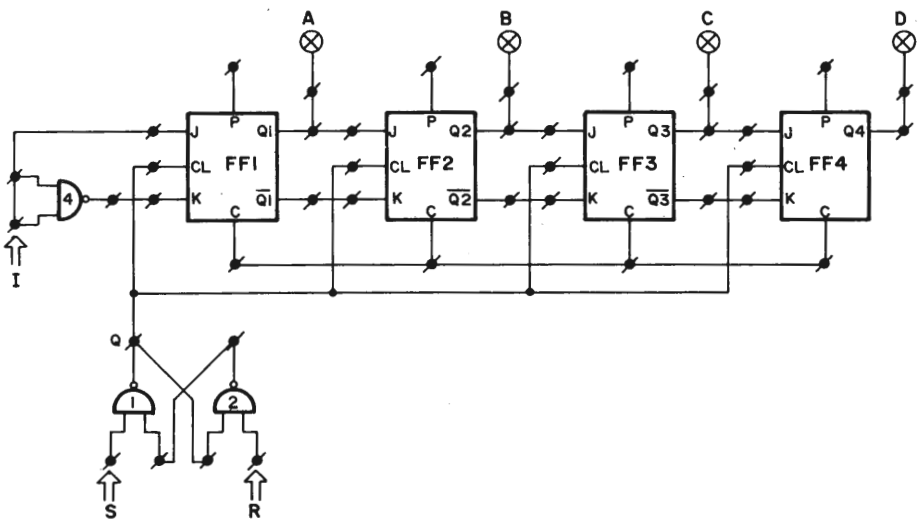


Fig. 6.1 *Schema di collegamento dell' esercitazione 1.*

re quattro impulsi di clock e osservare gli stati delle uscite. Otterrete la tabella di verità che segue:

Clock	A	B	C	D
0	1	0	0	1
2	1	1	0	0
3	0	0	1	1
4	1	0	0	1

III. Questa esercitazione dimostra che i dati immagazzinati in un registro ad anello non si perdono mai anche se si introducono un numero eccessivo di impulsi di clock.

ESERCITAZIONE 4. CONTATORE JOHNSON

I. Per realizzare un contatore Johnson basta realizzare il circuito di Fig. 6.1, rammentandosi di collegare l'uscita $\overline{Q}$ dell'ultimo bistabile all'ingresso segnato con I. Ricordarsi di inizializzare il contatore.

II. Applicare alcuni impulsi di clock e osservare lo stato delle uscite. Otterrete la tabella di verità che segue:

Clock	A	B	C	D
inizio	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

III. Il numero di stati di uscita di un contatore Johnson è esattamente il doppio rispetto ad un contatore ad anello con lo stesso numero di bit. Un vantaggio che ha rispetto al contatore ad anello è che parte automaticamente.

CAPITOLO 7

LA PIASTRA SPERIMENTALE

Tutti i circuiti logici utilizzati in questo equipaggiamento sperimentale, PIASTRA DIGITALE II, sono del tipo CMOS. Sono stati scelti circuiti di questa famiglia logica visto il costo ridotto e la facile acquisizione, qualità che condivide con i TTL, inoltre ha un maggior livello di integrazione, minor consumo e maggior immunità al rumore.

Nei prossimi argomenti che tratteremo, i circuiti logici verranno analizzati come "scatole nere" che contengono le funzioni logiche necessarie, senza riferirci alla loro circuiteria interna, dato che essa non è importante per la comprensione di dette funzioni.

7.1 Fonte di alimentazione

La piastra sperimentale richiede una fonte di alimentazione stabilizzata di 5 V e 400 mA. La si può ottenere attraverso un ponte raddrizzatore, filtri di condensatori e un regolatore di tensione, montati sulla scheda inferiore, delle tre che compongono l'equipaggiamento. L'unico collegamento esterno per l'alimentazione si esegue con il secondario di un trasformatore di 8 V e 500 mA, che verrà collegato ai terminali segnati per tale funzione nel circuito stampato.

Tuttavia, l'utilizzo della fonte di alimentazione non è indispensabile, coloro che hanno accesso ad una fonte di alimentazione di laboratorio di 5 V possono adoperarla senza problemi al posto del circuito di Fig. 7.1. In questo caso si omettono i componenti della fonte di alimentazione di detta figura. La fonte che si desidera utilizzare si collocherà in C2, rispettando la polarità. L'utilizzo di batterie non è consigliabile, dato che si esauriscono in poche ore a causa del consumo di corrente del circuito.

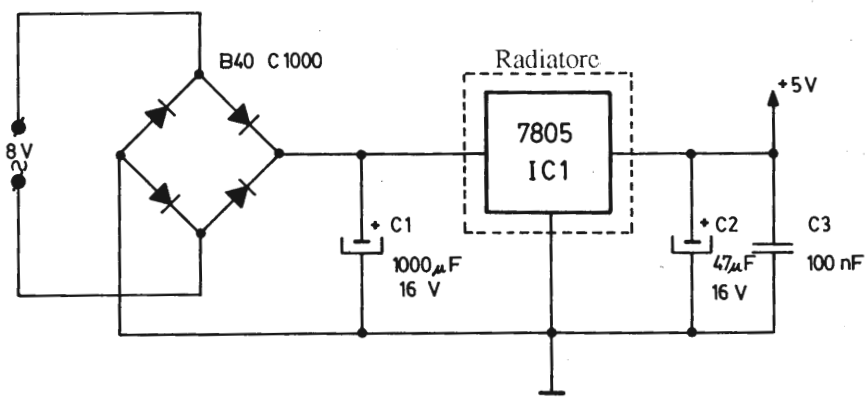


Fig. 7.1 *Fonte di alimentazione della piastra sperimentale.*

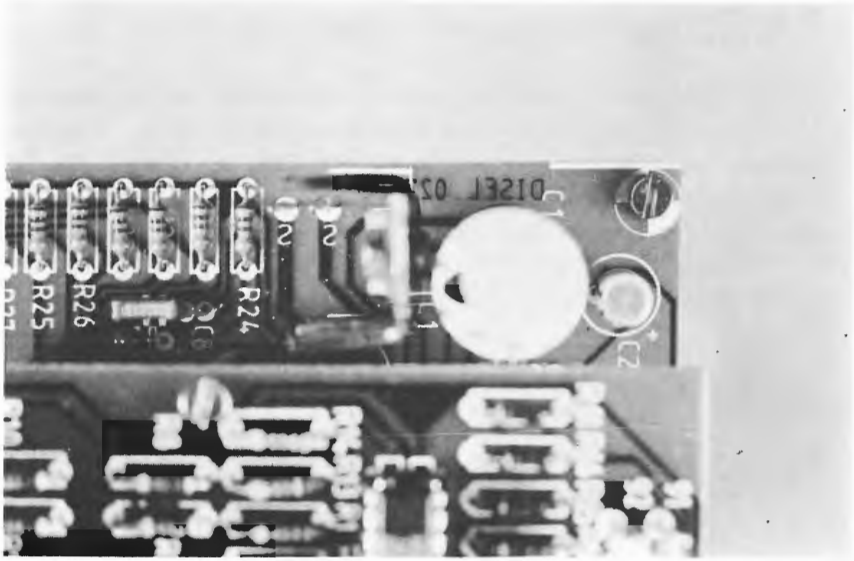


Fig. 7.2 *Particolare della fonte di alimentazione.*

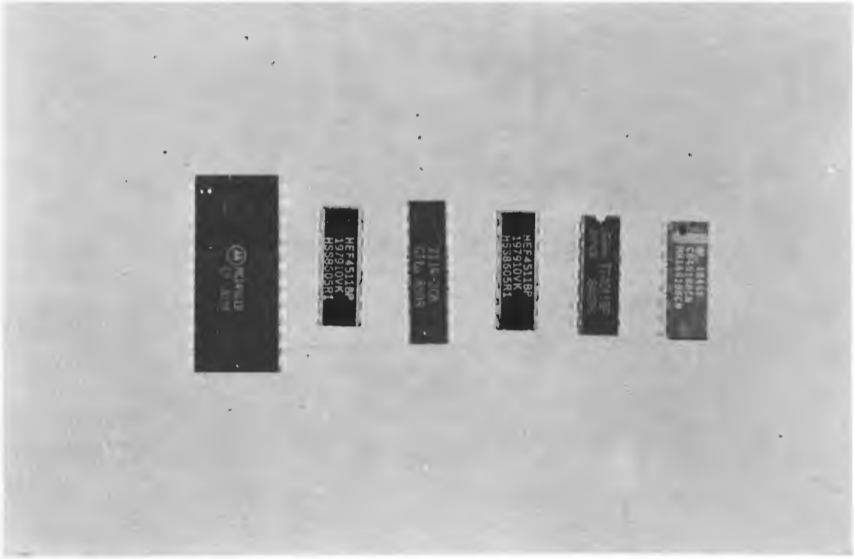


Fig. 7.3 *Circuiti integrati 4511, 4518, 4581, 4011 e 2114, la maggior parte di questi componenti sono utilizzati per eseguire le esercitazioni. Sono in logica CMOS.*

7.2 Componenti principali del circuito

I principali componenti dell'equipaggiamento sono installati sulle due schede inferiori. Nella prima, oltre alla fonte di alimentazione, troviamo i seguenti circuiti integrati logici: uno del tipo 4581, che è una ALU (unità aritmetico logica), questo integrato verrà trattato nel prossimo capitolo; uno del tipo 2114, memoria RAM di tipo statico, che verrà anch'essa descritta più avanti. Ven-

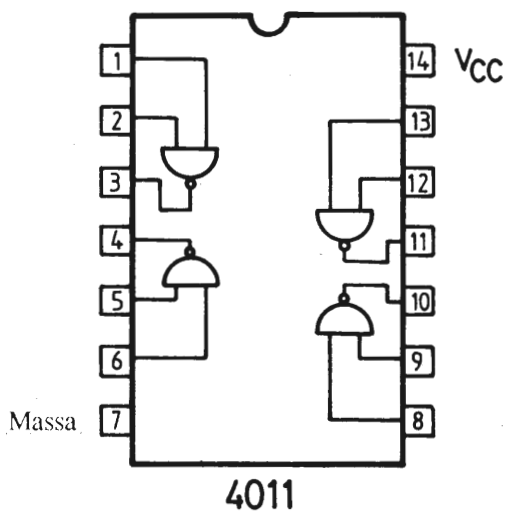


Fig. 7.4 *Piedinatura del circuito integrato, della famiglia CMOS, 4011; non è utilizzato nella piastra sperimentale ma è analogo al 7400.*

gono anche inclusi quattro LED con un transistor commutatore, collegato come mostrato in Fig. 7.5. Ogni LED è stato protetto per mezzo di una resistenza limitatrice di 330 Ω, che può essere ridotta a 120 Ω se si considera bassa l'intensità di luce emessa dal diodo.

Nella scheda posta in posizione mediana troviamo i seguenti circuiti integrati logici: uno del tipo 7400, che contiene 4 porte NAND a due ingressi; uno del tipo 4518, che è un contatore decimale doppio e due del tipo 4511, che so-

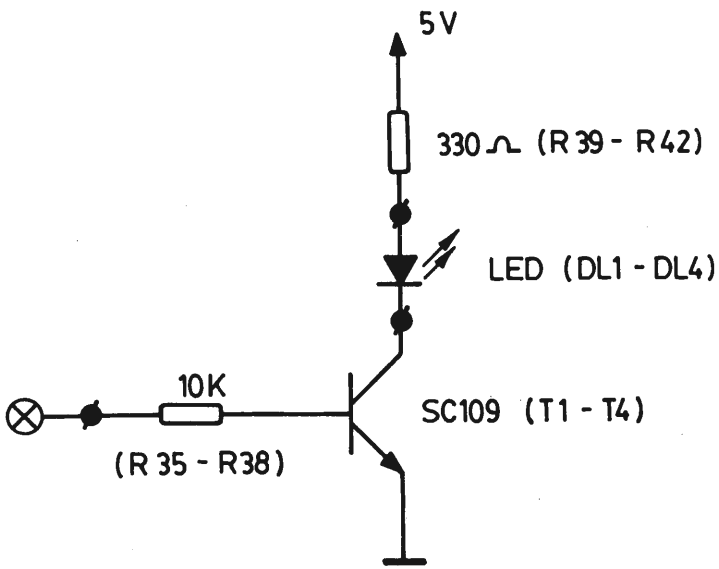


Fig. 7.5 *Circuito di eccitazione dei LED che vengono utilizzati per segnalare gli stati logici.*

no decodificatori per display a 7 segmenti. Con il circuito 7400, realizzate 2 bistabili del tipo RS, per ottenere impulsi di clock privi di rimbalzi, così come mostrato in Fig. 7.6. Questi bistabili vengono realizzati con componenti della famiglia TTL per evitare di fissare i livelli di entrata, indispensabile con componenti della famiglia CMOS.

Quando tutti i componenti sono stati montati e saldati sulle due schede inferiori, verranno introdotti e saldati i terminali di collegamento corrisponden-

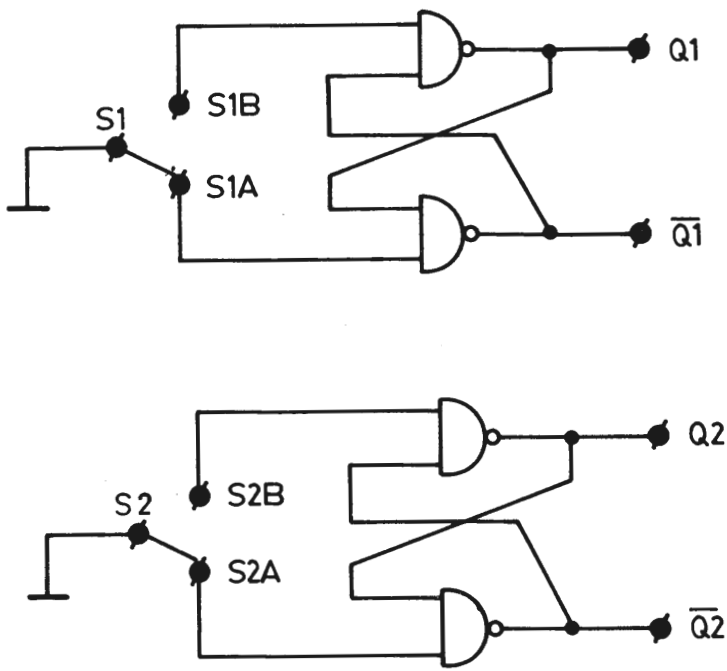


Fig. 7.6 *Montaggio di bistabili RS per ottenere, attraverso pulsanti, gli impulsi di clock necessari nelle esercitazioni.*

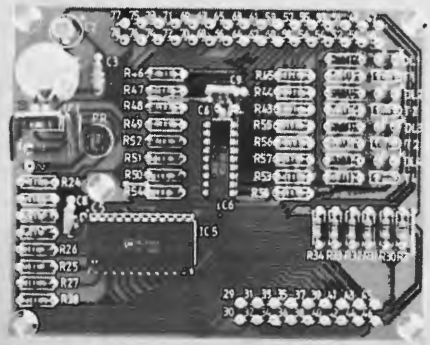


Fig. 7.7 *Scheda inferiore della piastra sperimentale con tutti i suoi componenti montati.*

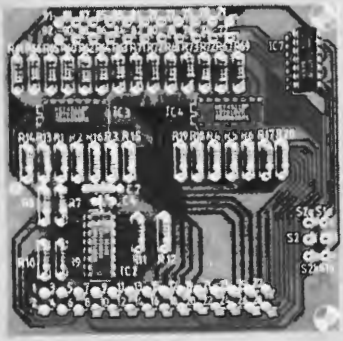


Fig. 7.8 *Scheda media della piastra sperimentale con tutti i suoi componenti montati.*

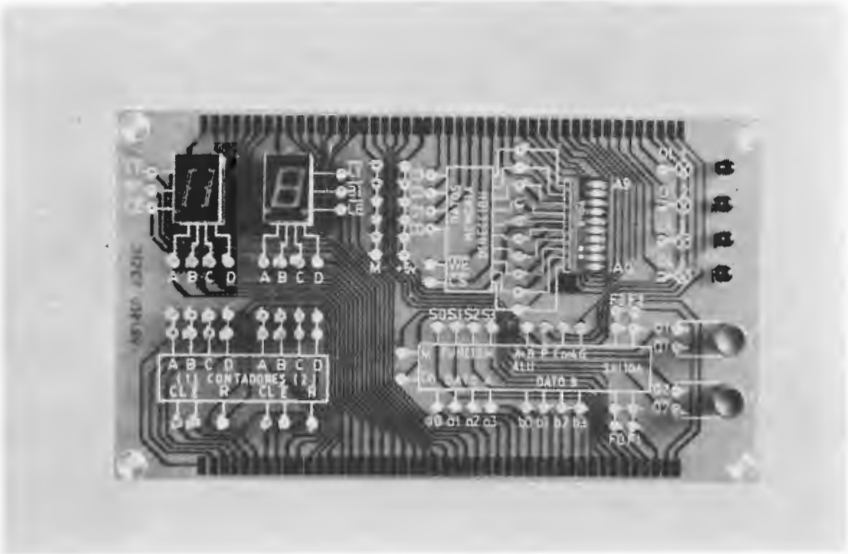


Fig. 7.9 *Scheda superiore della piastra sperimentale con tutti i suoi componenti montati. Sostituisce il coperchio della scatola.*

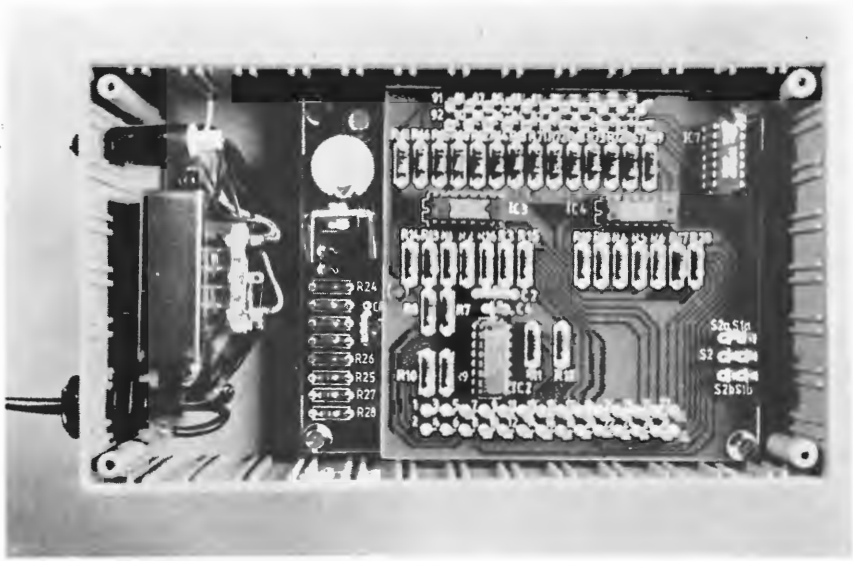


Fig. 7.10 *Disposizione dei diversi elementi della piastra sperimentale sulla scatola scelta.*

ti all'alimentazione in alternata di 8 V, quelli dei LED e quelli di collegamento tra le due schede inferiori e quella superiore.

Una volta completato il montaggio delle schede inferiori si procede in modo analogo con quella superiore, che contiene due display (visualizzatori a sette segmenti) del tipo HP7760 e un blocco di microinterruttori di 10 bit.

Il montaggio della scheda verrà ultimato saldando i terminali, i LED, i quali verranno attaccati con qualche goccia di adesivo plastico e i pulsanti; si può, a questo punto, passare all'ultima fase, che è il collegamento tra le tre schede. Il collegamento tra le due schede inferiori, si riduce esclusivamente all'alimentazione, mentre con la superiore lo si realizza direttamente attraverso cavo piatto, come viene mostrato in Fig. 7.11.

Sia il trasformatore che il collegamento alla rete, il portafusibile, la lampada spia e l'interruttore di messa in funzione saranno montati sulla scatola come meglio conviene. In Fig. 7.14 mostriamo a titolo orientativo il piano di foratura del frontale. La scheda superiore sostituisce il coperchio della scatola, di conseguenza è stato previsto di avvitarla nei fori predisposti per tale fine.

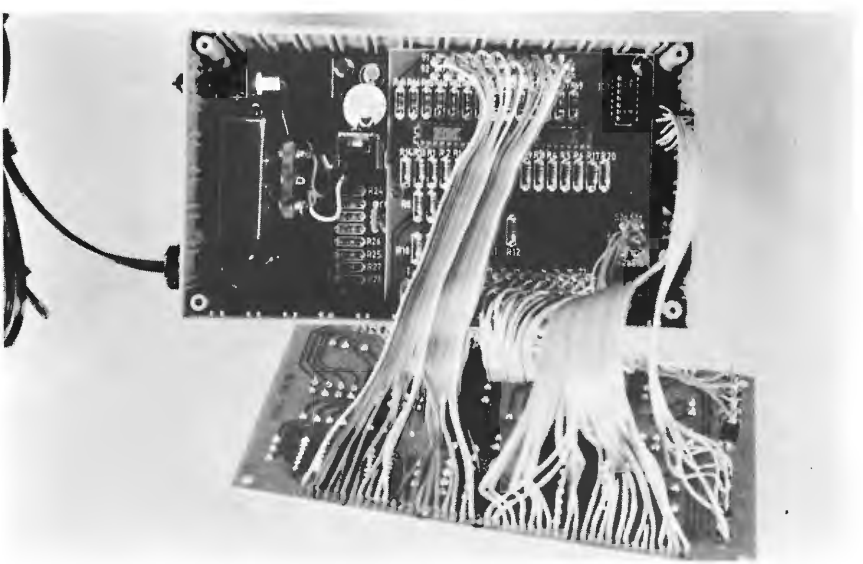


Fig. 7.11 *Particolare del collegamento tra le tre piastre, realizzato con cavo piatto.*

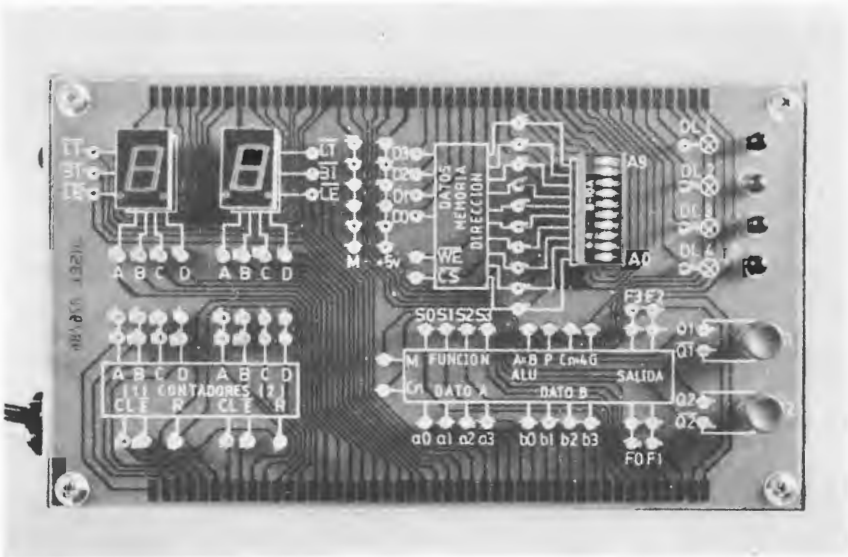


Fig. 7.12 *Aspetto finale della piastra sperimentale. Si osservi la serigrafia che ci indicherà nelle esercitazioni gli elementi che dobbiamo utilizzare.*

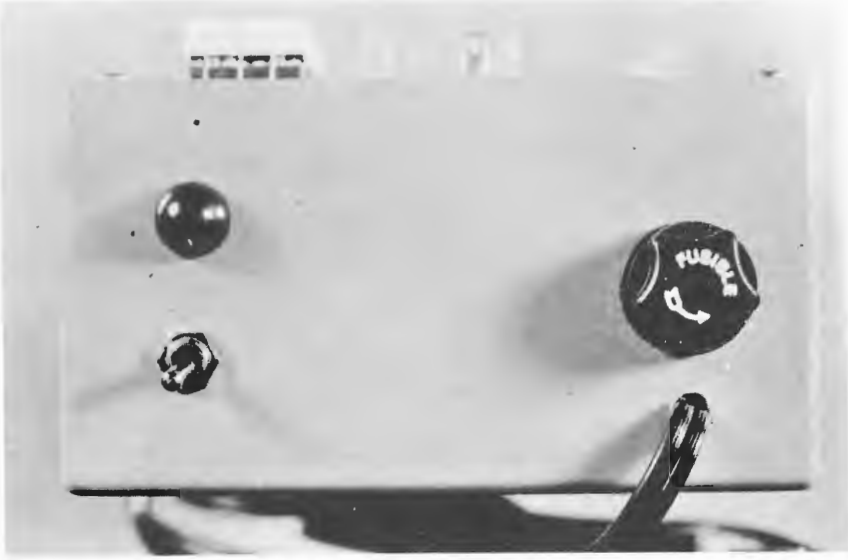


Fig. 7.13 *Particolare della piastra nel quale si può notare il portafusibili, la lampada spia, l'interruttore di rete e il cavo di collegamento a rete.*

7.3 Consigli per l'uso

Per facilitare il collegamento dei circuiti logici, le connessioni di ingresso e uscita si realizzeranno in terminali inseriti nel coperchio della piastra. Questi collegamenti si eseguiranno attraverso connettori faston con cavi di diverse lunghezze.

Esistono terminali di collegamento in vari punti, che corrispondono alle piste di alimentazione, + 5 V e massa, ai circuiti che controllano i LED (segnati con ⊗). Ogni componente è identificato con un nome in modo da essere facilmente utilizzato durante le esercitazioni pratiche.

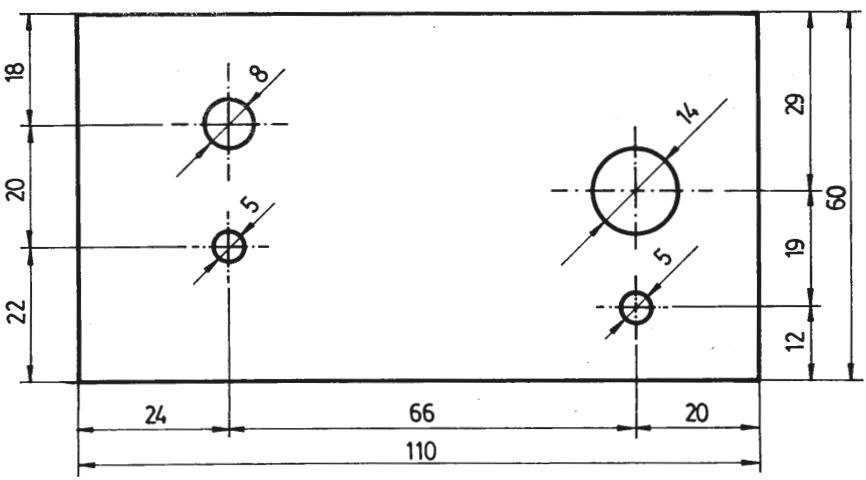


Fig. 7.14 *Piano di foratura del frontale della scatola.*

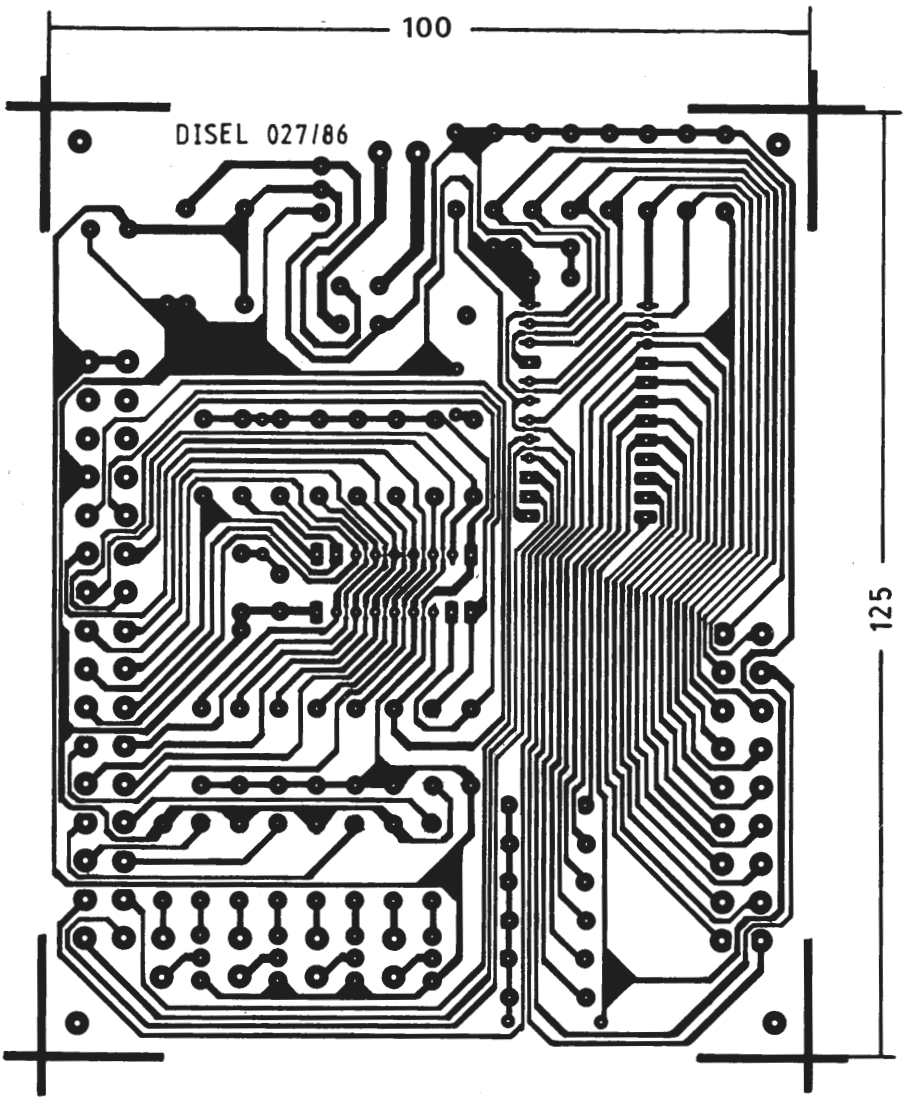


Fig. 7.15 *Facciata ramata della scheda inferiore, in scala 1:1.*

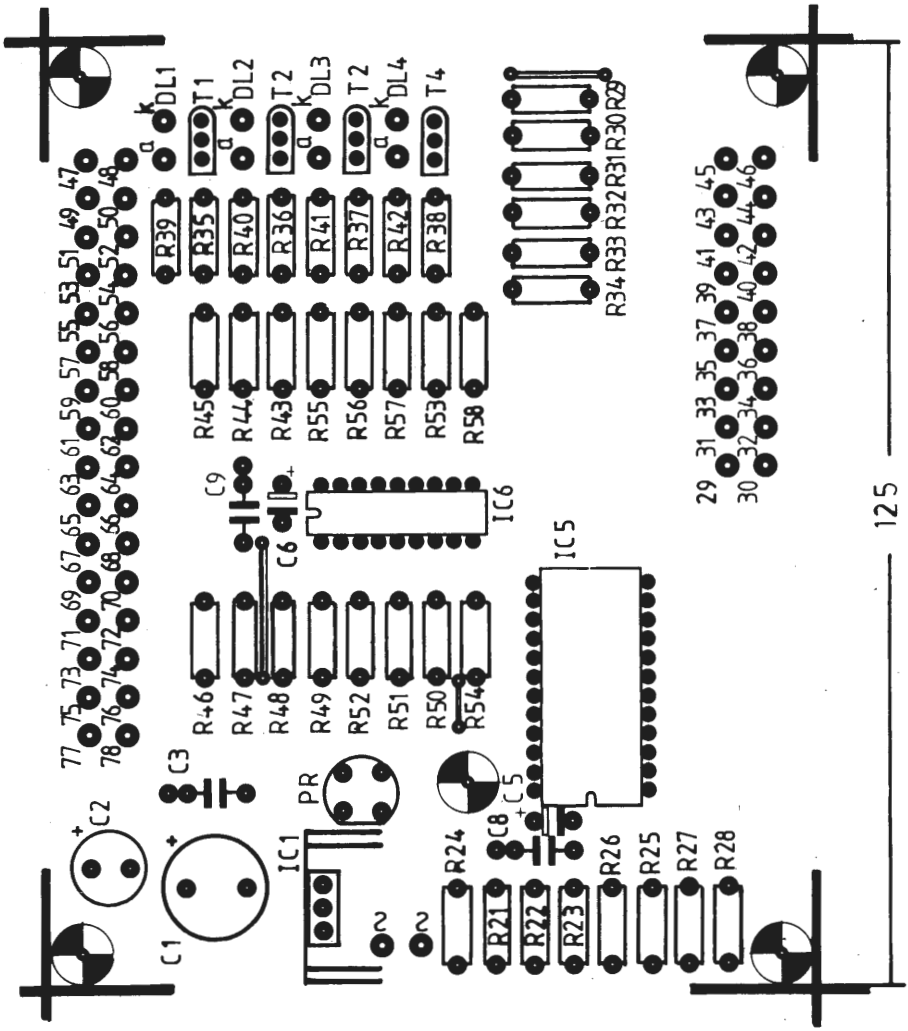


Fig. 7.16 *Serigrafia dei componenti della scheda inferiore.*

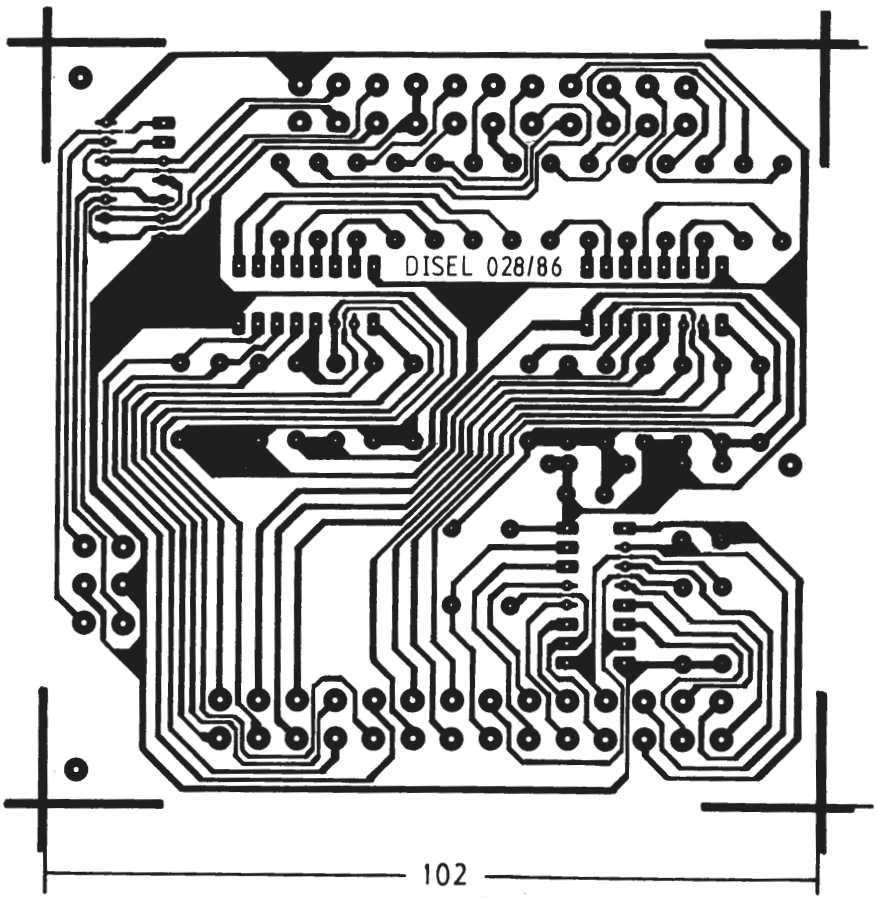


Fig. 7.17 *Facciata ramata della scheda media, in scala 1:1.*

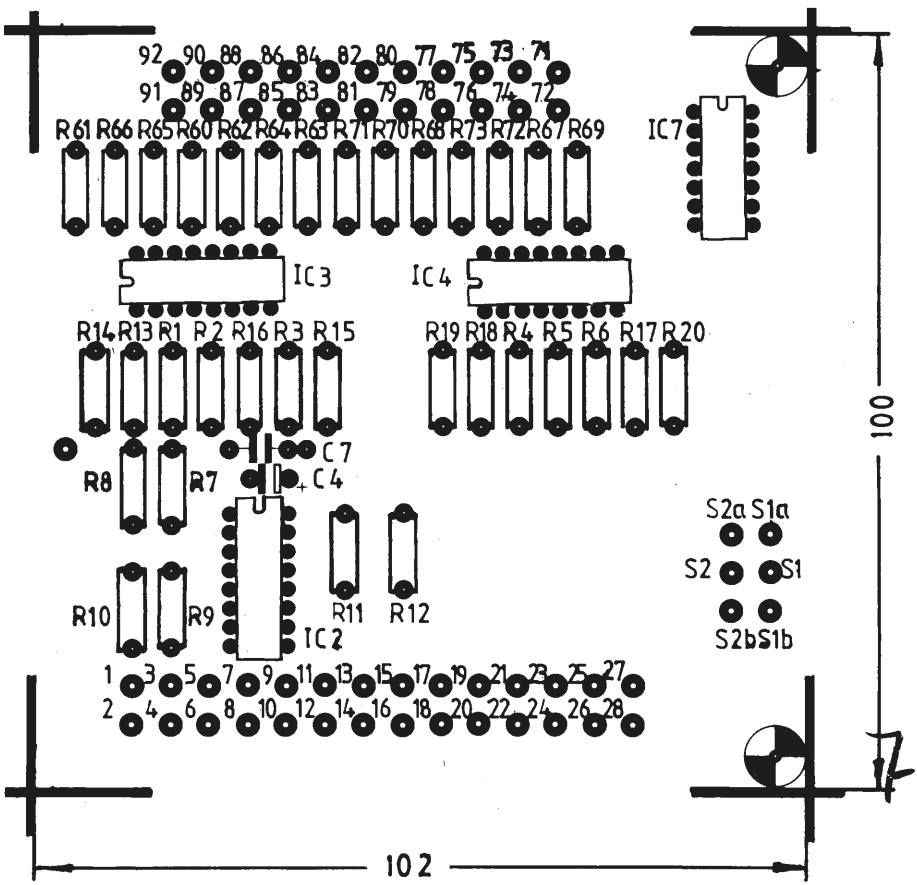


Fig. 7.18 *Serigrafia dei componenti della scheda media.*

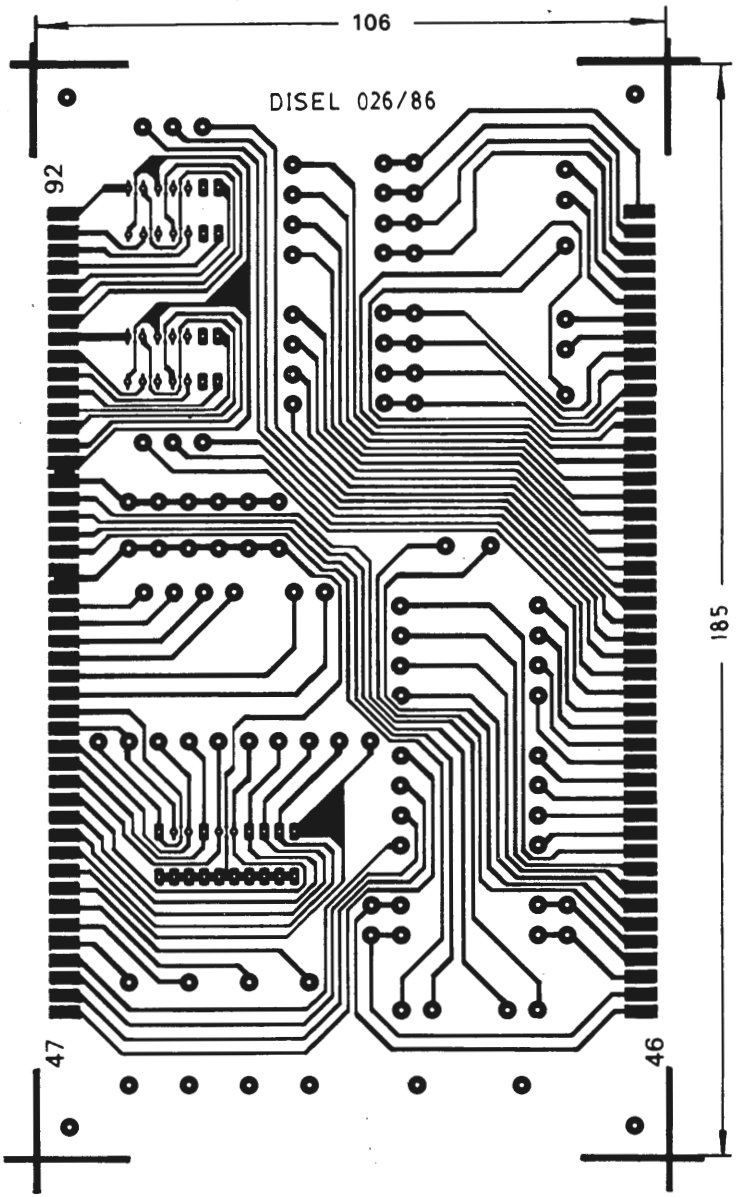


Fig. 7.19 *Facciata ramata della scheda superiore, in scala 1:1.*

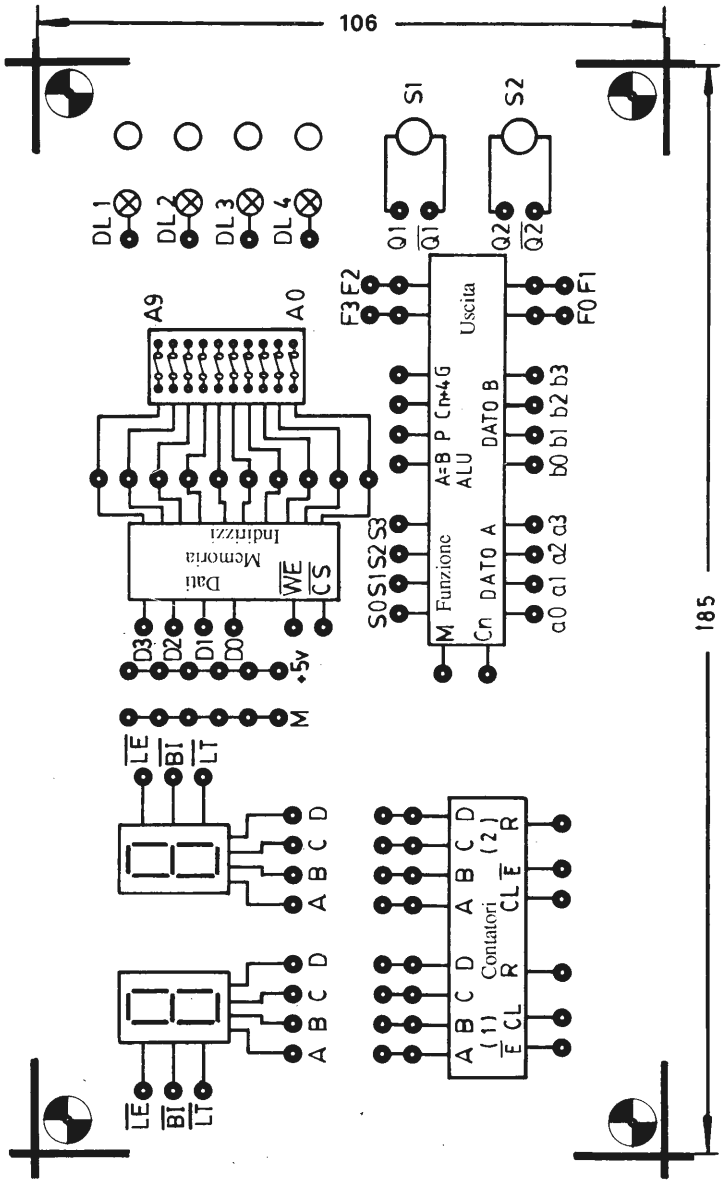


Fig. 7.20 *Serigrafia dei componenti della scheda superiore.*

Per evitare che il circuito stampato diventi una specie di riccio, nella maggior parte dei casi, è stato limitato a uno il numero di punti di connessione, tranne in qualche uscita dove ne sono stati messi due. Se bisogna fare più di un collegamento in un determinato terminale, si utilizzeranno cavi di connessione doppi, tripli o quadrupli.

Tutti i componenti utilizzati nella piastra sperimentale devono essere della miglior qualità, in particolare i circuiti integrati, che dovranno essere di una marca conosciuta e affidabile. anche se i circuiti integrati CMOS utilizzati sono molto affidabili, possono creare inconvenienti; per questo motivo raccomandiamo l'utilizzo di zoccoli, in modo tale da facilitare la sostituzione in caso di guasto.

7.4 Elenco dei componenti del preparatore digitale II

Resistenze: (tutte da 1/4 W)

- R1 a R20 di 100 kΩ (marrone, nero, giallo).
- R21 a R38 di 10 kΩ (marrone, nero, arancione).
- R39 a R42 di 330 Ω (arancione, arancione, nero).
- R43 a R59 di 2,2 kΩ (rosso, rosso, rosso).
- R60 a R73 di 330 Ω (arancione, arancione, nero).

Condensatori

- C1 elettrolitico da 1000 µF, 16 V, radiale.
- C2 elettrolitico da 47 µF, 16 V, radiale.
- C3, C7, C8 e C9 poliestere da 100 nF.
- C4, C5 e C6 tantalio da 1 µF, 16 V.

Semiconduttori

- IC1 7805.
- IC2 4518.
- IC3 e IC4 4511.
- IC5 4581.

- IC6 2114.
- IC7 7400.
- T1, T2, T3 e T4 SC109 o equivalente.
- DL1, DL2, DL3 e DL4 LED di 5 mm ∅.
- Ponte raddrizzatore B40 C1000.
- DP1 e DP2 display HP7760.

Varie

- 3 zoccoli per circuito integrato a 14 pin.
- 3 zoccoli per circuito integrato a 16 pin.
- 1 zoccolo per circuito integrato a 18 pin.
- 1 zoccolo per circuito integrato a 20 pin.
- 1 zoccolo per circuito integrato a 24 pin.
- 206 terminali.
- 50 connettori faston.
- 0,5 m. di cavo per collegamento.
- 1 radiatore piccolo per il regolatore.
- 2 separatori da 10 mm. da unire con altri e 2 viti.
- 1 separatore da 10 mm. e 2 viti.
- 4 separatori da 5 mm. e 4 viti.
- Trasformatore da 220 V / 8 V, 500 mA.
- Lampada spia da 220 V.
- Interruttore.
- Portafusibili.
- Fusibile da 500 mA.
- Viti e dadi per montare il trasformatore sulla scatola e il regolatore al radiatore.
- Un blocco di microinterruttori da 10 bit.
- 2 commutatori a pulsante.
- Scatola Polibox RP.4 GA o equivalente (190x110x160).
- 5 m. di cavo per collegamenti.
- Cavo per il collegamento alla rete.

CAPITOLO 8

CONTATORI INTEGRATI

Come già è stato detto nel capitolo precedente i contatori sono composti da diversi bistabili opportunamente collegati tra loro. Abbiamo studiato il modo di realizzare contatori decimali, ma grazie allo sviluppo tecnologico esistono contatori integrati in un solo chip (circuito integrato). Il funzionamento di un componente di questo tipo, che contiene due contatori decimali, è l'argomento che affronteremo in questo capitolo.

8.1 Contatore decimale

Da un divisore binario di 1:16 si può ottenere un divisore decimale 1:10, se si resetta il divisore completo quando arriva al numero 10, come visto in precedenza. Spesso i divisori decimali sono composti da divisori binari 1:2 e 1:5.

Il divisore 1:5 si realizza, in questo caso, attraverso un divisore 1:8, il quale si resetta dopo il 5º impulso.

Andiamo ora ad analizzare più in concreto il circuito integrato 4518, appartenente alla media scala di integrazione (MSI) della serie CMOS, che utilizzeremo nelle nostre esercitazioni.

Questo componente contiene due contatori indipendenti di tipo BCD (decimale codificato in binario).

Le diverse configurazioni che possono assumere le uscite del contatore sono fornite da quattro flip-flop di tipo D. Il clock e la linea di abilitazione (enable) sono intercambiabili. Guardando la tabella di verità, si nota infatti, che il contatore può essere incrementato sul fronte di salita dell'impulso di clock, con enable a 1, oppure sul fronte di discesa dell'enable, con clock a 0. Ogni contatore può essere azzerato in modo indipendente, applicando un livello alto all'ingresso di reset.

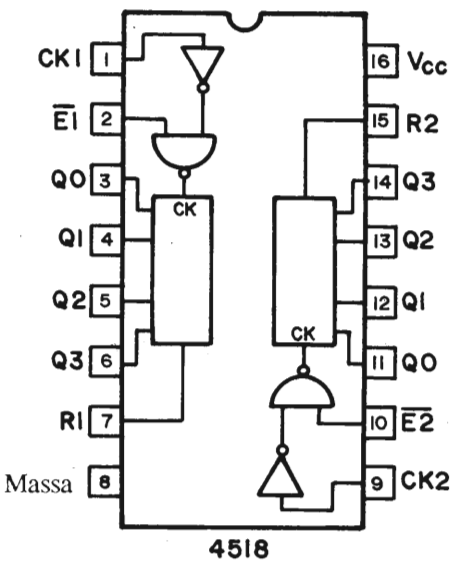


Fig. 8.1 *Schema a blocchi e piedinatura del circuito integrato 4518.*

Di seguito riportiamo la tabella di verità del circuito integrato 4518:

Clock	Enable	Reset	Azione
↗	1	0	Incrementa il contatore
0	↘	0	Incrementa il contatore
↘	X	0	Non cambia
X	↗	0	Non cambia
↗	0	0	Non cambia
1	↘	0	Non cambia
X	X	1	Q0 a Q3 = 0

X = Indifferente

8.2 Codici utilizzati per la visualizzazione

I più usati sono il codice decimale e quello a sette segmenti. Per esprimere un numero in codice decimale servono 10 bit; ogni configurazione è caratterizzata da un bit a valore 1 e dai rimanenti 9 bit a valore 0.

	0	1	2	3	4	5	6	7	8	9
0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1

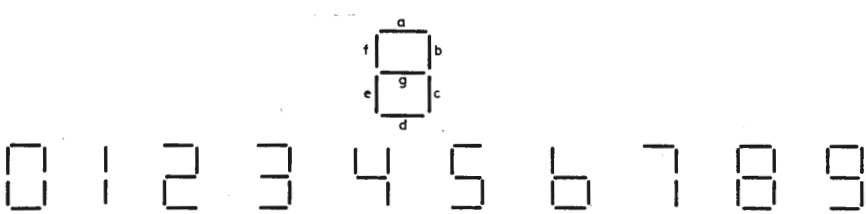


Fig. 8.2 *Codice a 7 segmenti.*

Il codice a sette segmenti è basato sulla proprietà che hanno sette segmenti, disposti e alimentati in modo opportuno, di rappresentare tutti i numeri decimali, come mostrato in Fig. 8.2.

Di conseguenza questo codice è:

	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	0	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

8.3 Decodificatori

I decodificatori sono circuiti combinatori che svolgono la funzione di operare una traduzione da un codice ad un altro. Sono costruiti, di conseguenza, con porte logiche, utilizzando le espressioni che permettono il passaggio da un codice all'altro. Questi circuiti sono molto utilizzati in applicazioni quali: il multiplexaggio di dati, il pilotaggio di display, convertitori digitali-analogici, indirizzamento di memorie, ecc.

Come esempio, vedremo l'esecuzione di una decodifica da BCD a sette segmenti, molto utilizzata in visualizzatori numerici.

La tabella di verità è:

D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	0	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1

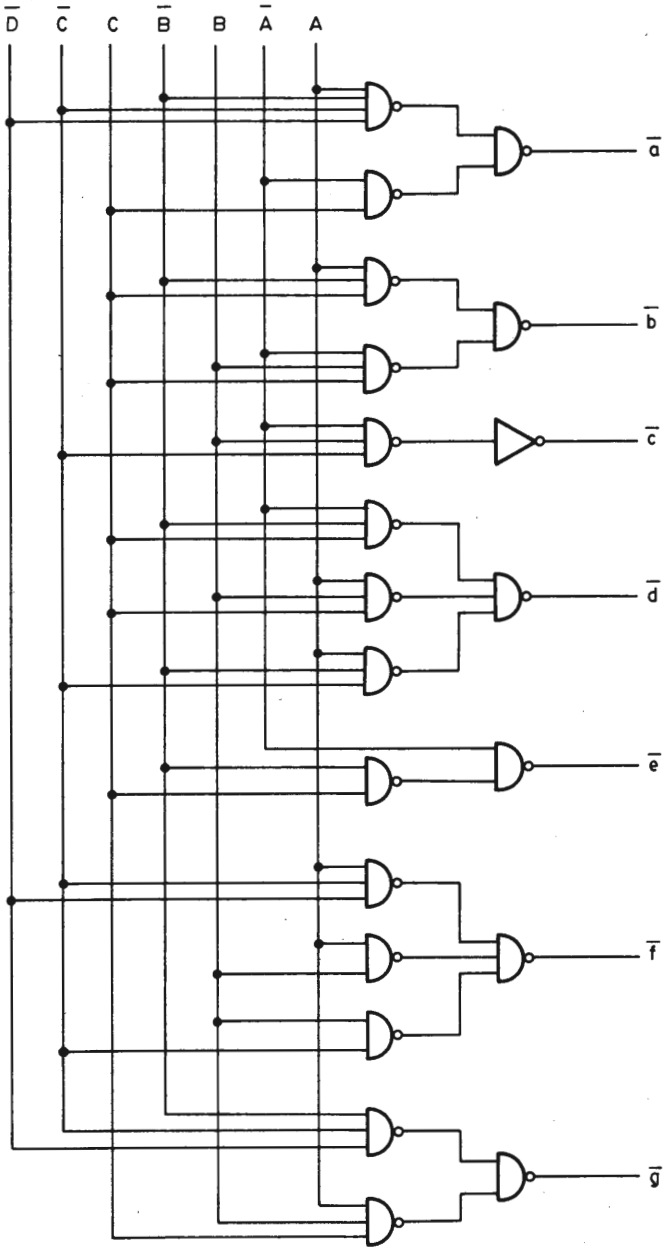


Fig. 8.3 *Decodificatore da BCD a 7 segmenti realizzato con porte logiche.*

Ricaviamo le corrispondenti mappe di Karnaugh:

Segmento a

DC \ BA	00	01	11	10
00	1	0	1	1
01	0	1	1	0
11	X	X	X	X
10	1	1	X	X

$$\overline{a} = A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot C$$

Segmento b

DC \ BA	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	X	X	X	X
10	1	1	X	X

$$\overline{b} = A \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot C$$

Segmento c

DC \ BA	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	X	X	X	X
10	1	1	X	X

$$\overline{c} = \overline{A} \cdot B \cdot \overline{C}$$

Segmento d

DC \ BA	00	01	11	10
00	1	0	1	1
01	0	1	0	1
11	X	X	X	X
10	1	0	X	X

$$\overline{d} = \overline{A} \cdot \overline{B} \cdot C + A \cdot B \cdot C + A \cdot \overline{B} \cdot \overline{C}$$

Segmento e

DC \ BA	00	01	11	10
00	1	0	0	1
01	0	0	0	1
11	X	X	X	X
10	1	0	X	X

$$\overline{e} = A + \overline{B} \cdot C$$

Segmento f

DC \ BA	00	01	11	10
00	1	0	0	0
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

$$\overline{f} = A \cdot \overline{C} \cdot \overline{D} + A \cdot B + B \cdot \overline{C}$$

DC \ BA	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

Segmento g

$$\overline{g} = \overline{B} \cdot \overline{C} \cdot \overline{D} + A \cdot B \cdot C$$

Con le equazioni ottenute, utilizzando solo porte NAND e una porta NOT si ottiene il circuito riportato in Fig. 8.3

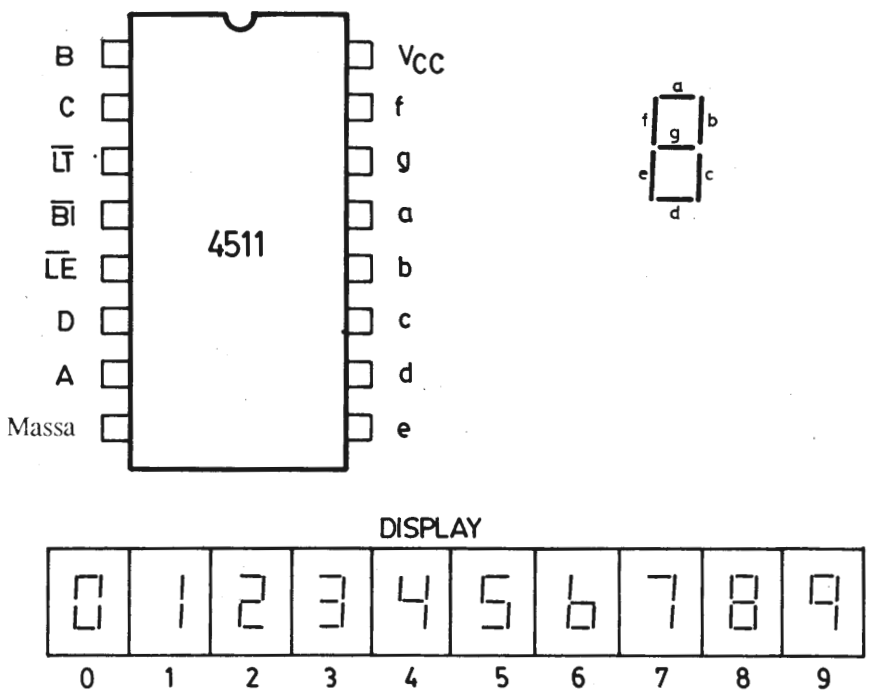


Fig. 8.4 *Piedinatura del circuito integrato della famiglia CMOS 4511.*

8.4 Decodificatore da BCD a 7 segmenti

Il circuito integrato 4511, della famiglia CMOS, è un decodificatore da BCD a sette segmenti, circuito della media scala di integrazione (MSI) possiede internamente uno stadio amplificatore che permette di pilotare i LED che compongono i 7 segmenti del display. Le uscite sono provviste di latch che permettono, volendo, la memorizzazione del dato. E' il decodificatore che utilizzeremo nelle nostre esercitazioni per collegare i display.

In Fig. 8.4 è riportata la piedinatura di questo componente, di seguito è riportata la tabella di verità:

ENTRATE							USCITE							
$\overline{LE}$	$\overline{BI}$	$\overline{LT}$	D	C	B	A	a	b	c	d	e	f	g	Visualizzatore
X	X	0	X	X	X	X	1	1	1	1	1	1	1	8
X	0	1	X	X	X	X	0	0	0	0	0	0	0	Bianco
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	1	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	Bianco
0	1	1	1	0	1	1	0	0	0	0	0	0	0	Bianco
0	1	1	1	1	0	0	0	0	0	0	0	0	0	Bianco
0	1	1	1	1	0	1	0	0	0	0	0	0	0	Bianco
0	1	1	1	1	1	0	0	0	0	0	0	0	0	Bianco
0	1	1	1	1	1	1	0	0	0	0	0	0	0	Bianco
1	1	1	X	X	X	X	*							*

X = Indifferente

* Dipende dal codice BCD preventivamente utilizzato quando LE=0.

Il pin $\overline{LT}$ serve per verificare il buon funzionamento dei 7 segmenti del display (lamp test), mettendolo a livello logico 0. Un altro pin di questo integra-

to è $\overline{BI}$ (blanking imput) che serve a spegnere tutti i segmenti del display. Infine il pin $\overline{LE}$ (latch enable) memorizza il dato presente sul display quando si invia un 1 a tale pin.

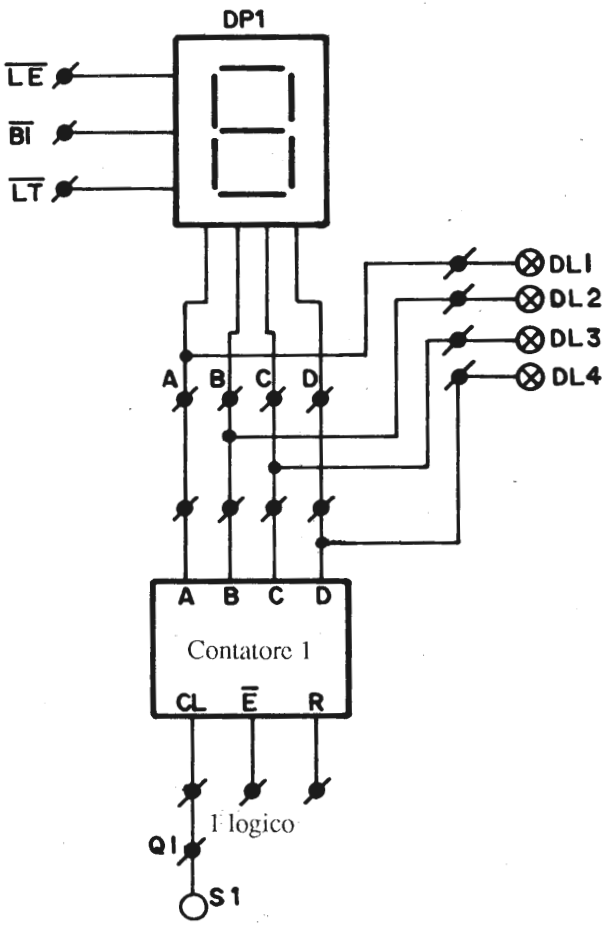


Fig. 8.5 *Schema di collegamento dell' esercitazione 1.*

Quando si invia un livello basso al pin $\overline{LE}$ il contenuto del latch (memoria tampone) viene visualizzato sul display.

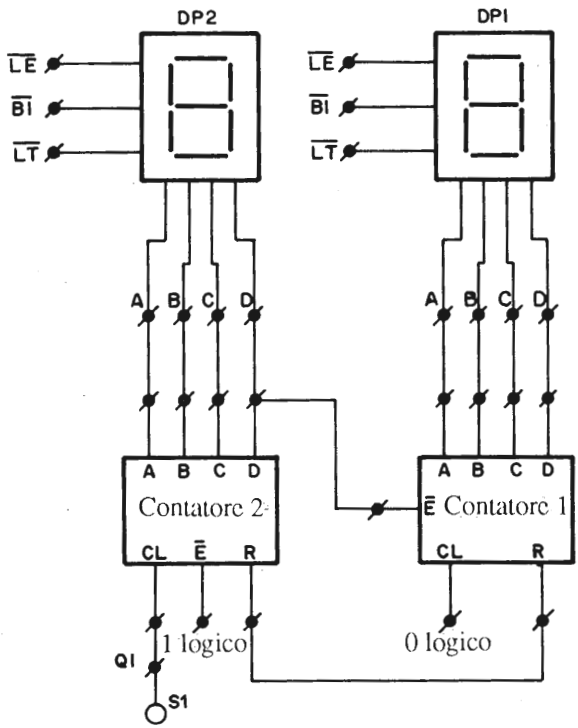


Fig. 8.6 *Schema di collegamento dell' esercitazione 3.*

ESERCITAZIONE 1. CONTATORE ATTIVO SUL FRONTE DI SALITA

I. Montare il circuito di Fig. 8.5

II. Quando verrà applicata la tensione di alimentazione il contatore assumerà una configurazione qualunque, che dipenderà dalle caratteristiche interne dello stesso. Questo stato arbitrario deve essere cancellato mettendo momentaneamente a zero l'entrata di reset (R).

III. Utilizzare il pulsante S1 per inviare impulsi all'ingresso di clock, tenendo presente che ogni volta che premiamo il pulsante inviamo impulsi di clock attivi sul fronte di salita. Si osserverà infatti che quando pigiamo il pulsante, il visualizzatore DP1 incrementerà il conteggio, mentre quando lo rilasciamo, non succederà nulla.

IV. Osservare per mezzo dei diodi LED, i diversi stati attraverso i quali passa il contatore. Confrontandoli con i numeri visualizzati dal display si otterrà la seguente relazione:

A	B	C	D	a	b	c	d	e	f	g	Display
0	0	0	0	1	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	0	0	1	1	9

V. Verificare tutte le possibili combinazioni della tabella di verità dell'integrato 4511.

ESERCITAZIONE 2. CONTATORE ATTIVO SUL FRONTE DI DISCESA

I. Montare il circuito di Fig. 8.5 ponendo il pin CL a livello logico 0 e introducendo gli impulsi di clock attraverso il pin $\overline{E}$.

II. Azzerare il contatore.

III. Utilizzando il pulsante S1 inviare gli impulsi di clock. Potrete osservare che il contatore non cambia quando premiamo il pulsante, ma quando lo lasciamo. Questo dipende dal fatto che adesso il contatore conta sul fronte di discesa.

ESERCITAZIONE 3. CONTATORI IN CASCATA

I. In alcune applicazioni è necessario poter contare più di un digit. Montando il circuito di Fig. 8.6 otterremo un contatore da 0 a 99, con un segnale di clock attivo sul fronte di salita.

II. Inviando impulsi attraverso il pulsante S1 incrementeremo il conteggio. E' conveniente azzerare i contatori prima di compiere qualsiasi operazione.

III. Se abbiamo la pazienza di introdurre più di 99 impulsi, potremo osservare che quando il contatore supera il numero 99 ricomincia a contare a partire dal numero 00.

CAPITOLO 9

UNITA' ARITMETICO-LOGICA

Un'unità aritmetico-logica è un circuito integrato molto complesso. Gli ingressi sono caratterizzati da un'entrata dati di due parole e un'uscita che rappresenta la funzione realizzata. Il circuito effettua, con i dati presenti in ingresso, la funzione indicata attraverso alcune entrate di selezione di funzione e attraverso un'altra entrata si seleziona la funzione aritmetica o logica.

9.1 Circuito integrato 4581

Tale circuito è un'unità aritmetico logica realizzata in tecnologia CMOS. Ha un'elevata immunità al rumore e un consumo molto ridotto, entrambe queste caratteristiche sono tipiche dei circuiti CMOS standard.

Questo integrato ha 75 porte integrate in un unico chip. Permette di eseguire 16 operazioni binarie di tipo aritmetico e logico con due parole di quattro bit. Tali operazioni sono selezionate attraverso quattro linee (S0, S1, S2 e S3), realizzando somme, sottrazioni, decrementi e spostamenti.

Per le operazioni aritmetiche bisogna applicare un livello basso all'entrata di controllo (M).

Quando viene utilizzato insieme all'integrato 4582 si possono eseguire operazioni aritmetiche di otto bit ad alta velocità.

Questo circuito ci permette di trattare dati attivi alti oppure bassi, a seconda della scelta dell'utente. Si può anche utilizzare come comparatore di quattro bit, dato che quando sono uguali l'uscita di uguaglianza (A = B) va a livello logico 1.

Per le operazioni logiche con due variabili, l'entrata di controllo (M) deve essere messa a livello alto.

E' in grado di svolgere le funzioni AND, NAND, OR, e NOR.

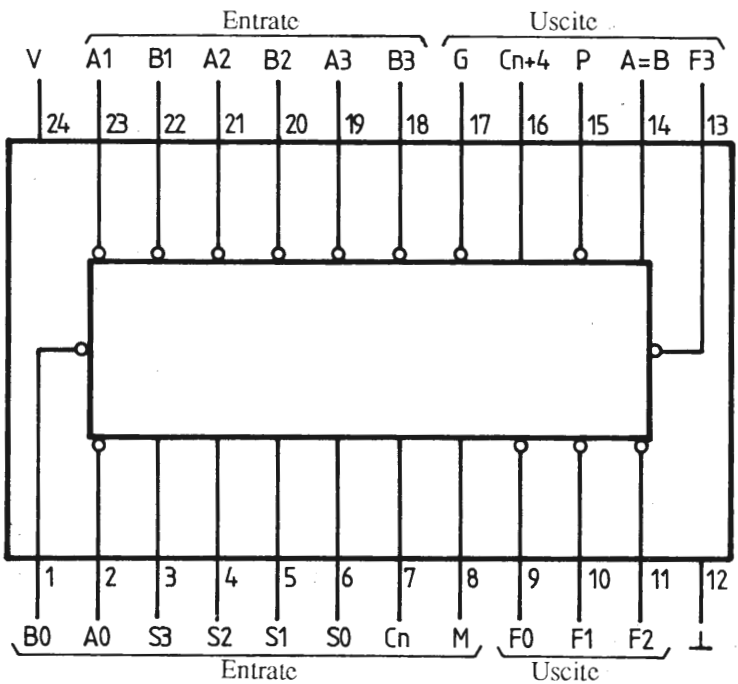


Fig. 9.1 *Piedinatura del circuito integrato 4518.*

Caratteristiche

- Dal punto di vista funzionale e della piedinatura è equivalente al 7481 della famiglia TTL.
- Bassa corrente di entrata: 5 nA a 5 V.
- Alta immunità al rumore: 45% della tensione di alimentazione.
- Tutte le entrate protette da diodi.
- Bassa capacità in entrata: 5 pF.
- Corrente d'uscita: 10 mA.

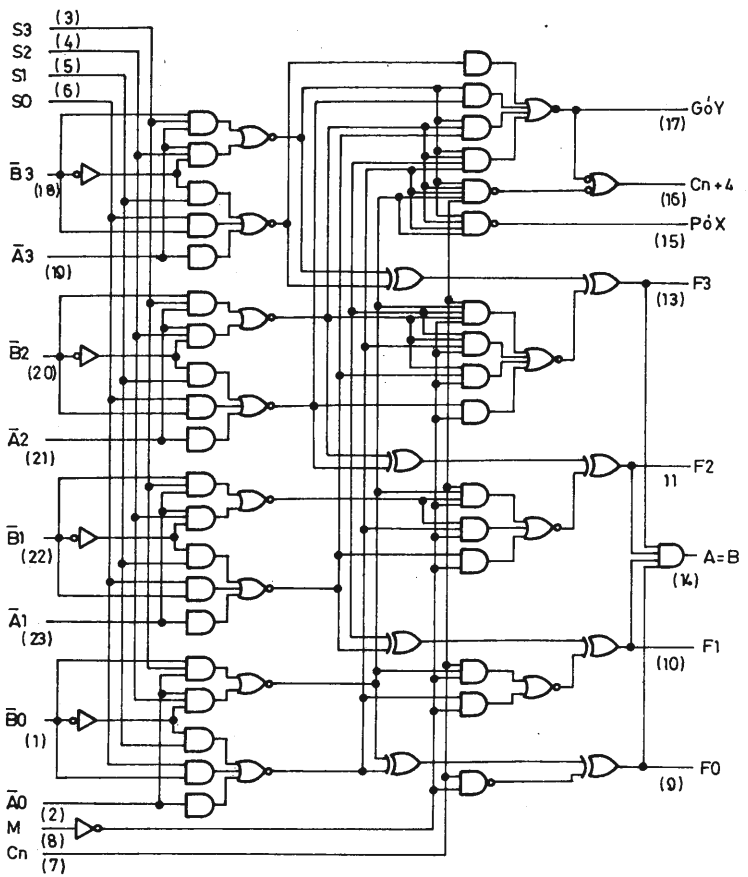


Fig. 9.2 *Circuito equivalente dell'integrato 4518.*

- Elevato range di alimentazione: da 3 a 8 V.
- Temperatura di immagazzinamento: da -65 a + 150 °C.
- Temperatura di funzionamento: da -40 a + 85 °C.

Le applicazioni più comuni di questo integrato si rivolgono alla soluzione di operazioni aritmetiche e logiche, raggiungendo, come si mostra nelle Figg. 9.3 e 9.4, alte prestazioni in combinazione con il 4582.

Di seguito esponiamo una serie di tabelle che mostrano: il primo la funzione svolta dai diversi piedini dell'integrato, il successivo la relazione esistente tra l'entrata C_n, l'uscita C_{n+4} e le due parole dati dell'entrata, infine la tabella di selezione per le diverse funzioni aritmetiche e logiche con riporto e senza riporto in logica positiva, cioè con dati attivi a livello alto.

Designazione	**Piedini**	**Funzione**
A3, A2, A1, A0	19, 21, 23, 2	Entrata parola A
B3, B2, B1, B0	18, 20, 22, 1	Entrata parola B
S3, S2, S1, S0	3, 4, 5, 6	Entrate di selezione di funzione
C_n	7	Entrata di riporto (carry)
M	8	Entrata di controllo del modo
F3, F2, F1, F0	13, 11, 10, 9	Funzione di uscita
A = B	14	Comparazione
P	15	Uscita di propagazione del riporto
C_{n+4}	16	Uscita di riporto
G	17	Uscita di riporto generata
V_{cc}	24	Alimentazione
⊥	12	Massa

Entrata C_n	**Uscita** C_{n+4}	**Dato attivo alto**	**Dato attivo basso**
1	1	$A \leq B$	$A \geq B$
1	0	$A > B$	$A < B$
0	1	$A < B$	$A > B$
0	0	$A \geq B$	$A \leq B$

					Dati attivi alti M = 0; operazioni aritmetiche	
S3	S2	S1	S0	M = 1 Funzioni logiche	$C_n = 1$ (senza riporto)	$C_n = 0$ (con riporto)
0	0	0	0	$F = \overline{A}$	$F = A$	$F = A + 1$
0	0	0	1	$F = \overline{A} + \overline{B}$	$F = A + B$	$F = A + B + 1$
0	0	1	0	$F = \overline{A}B$	$F = A + \overline{B}$	$F = A + \overline{B} - 1$
0	0	1	1	$F = 0$	$F = -1$	$F = 0$
0	1	0	0	$F = \overline{A}\,\overline{B}$	$F = A + A\overline{B}$	$F = A + A\overline{B} + 1$
0	1	0	1	$F = \overline{B}$	$F = A + B + A\overline{B}$	$F = A + B + A\overline{B} + 1$
0	1	1	0	$F = A \oplus B$	$F = A - B - 1$	$F = A - B$
0	1	1	1	$F = A\overline{B}$	$F = A\overline{B} - 1$	$F = A\overline{B}$
1	0	0	0	$F = \overline{A} + B$	$F = A + AB$	$F = A + AB + 1$
1	0	0	1	$F = \overline{A \oplus B}$	$F = A + B$	$F = A + B + 1$
1	0	1	0	$F = B$	$F = A + \overline{B} + AB$	$F = A + \overline{B} + AB + 1$
1	0	1	1	$F = AB$	$F = AB - 1$	$F = AB$
1	1	0	0	$F = 1$	$F = A + A$	$F = A + A + 1$
1	1	0	1	$F = A + \overline{B}$	$F = A + B + A$	$F = A + B + A + 1$
1	1	1	0	$F = A + B$	$F = A + \overline{B} + A$	$F = A + \overline{B} + A + 1$
1	1	1	1	$F = A$	$F = A - 1$	$F = A$

ESERCITAZIONE 1. OPERAZIONI LOGICHE

I. Si osservi lo schema di Fig. 9.5. Si indicano soltanto i collegamenti dei segnali a livello logico 1, poichè gli altri sono sicuramente posti a zero. Questo si ottiene ponendo gli ingressi a massa attraverso una resistenza di 10 KΩ (da R21 a R43), già installate sul circuito stampato inferiore. Verificare, utilizzando i diodi LED, prima di montare il circuito. Internamente, il complesso ha i segnali del decodificatore da BCD a 7 segmenti, nei livelli adeguati per visualizzare il risultato delle operazioni che di seguito si realizzeranno.

II. La prima cosa che bisogna fare per eseguire operazioni logiche con parole di quattro bit, è mettere a 1 l'entrata di controllo (M). In seguito verrà selezionata la combinazione adeguata utilizzando le entrate S0, S1, S2 e S3. La combinazione riportata nello schema ci permette di realizzare una funzione OR.

III. L'esempio che tratteremo assume il numero binario 0011 come dato A e 0010 come dato B, considerando dati e segnali attivi a livello alto.

Svolgendo tutte le possibili funzioni logiche otterremo i seguenti risultati:

M = 1 ; DATO A = 0011 ; DATO B = 0010						
S3	S2	S1	S0	Funzione	Uscita	Display
0	0	0	0	$F = \overline{A}$	1100	bianco
0	0	0	1	$F = \overline{A} + \overline{B}$	0011	3
0	0	1	0	$F = \overline{A}B$	0000	0
0	0	1	1	$F = 0$	0000	0
0	1	0	0	$F = \overline{A}\,\overline{B}$	1100	bianco
0	1	0	1	$F = \overline{B}$	1101	bianco
0	1	1	0	$F = A \oplus B$	1101	bianco
0	1	1	1	$F = A\overline{B}$	0001	1
1	0	0	0	$F = \overline{A} + B$	1110	bianco
1	0	0	1	$F = \overline{A \oplus B}$	0010	2
1	0	1	0	$F = B$	0010	2
1	0	1	1	$F = AB$	0010	2
1	1	0	0	$F = 1$	1111	bianco
1	1	0	1	$F = A + \overline{B}$	1111	bianco
1	1	1	0	$F = A + B$	0011	3
1	1	1	1	$F = A$	0011	3

IV. Come si può vedere dai risultati riportati in tabella, con questa ALU possiamo realizzare praticamente tutte le operazioni logiche esistenti con due parole di 4 bit. È conveniente ripetere la prova precedente assegnando valori diversi sia al dato A che al dato B. Proponiamo di farlo con DATO A=0101 e DATO B=1010.

V. Vediamo come funziona l'entrata C_n, le uscite C_{n+4} e A = B in funzione dei nostri due esempi. Modificare il collegamento ai LED portandoci le uscite C_{n+4} e A = B, in modo tale da visualizzarne i valori. Otterremo i seguenti risultati.

DATO A = 0011 ; DATO B = 0010		
C_n	C_{n+4}	A = B
0	0	0
1	0	0
DATO A = 0101 ; DATO B = 1010		
C_n	C_{n+4}	A = B
0	1	0
1	1	0

Il segnale A = B andrà a livello logico 1 soltanto quando il DATO A è uguale al DATO B. Verificare i risultati in base alla relazione esistente tra C_n, C_{n+4} e le due parole dati precedenti.

ESERCITAZIONE 2. OPERAZIONI ARITMETICHE

I. Questo paragrafo è identico a quello dell'esercitazione precedente.

II. La prima cosa che bisogna fare per realizzare operazioni aritmetiche con parole di quattro bit, è mettere a zero l'ingresso di controllo (M). In seguito verrà selezionata la combinazione adeguata utilizzando gli ingressi S0, S1, S2 e S3. La combinazione riportata nello schema ci permette di eseguire una somma.

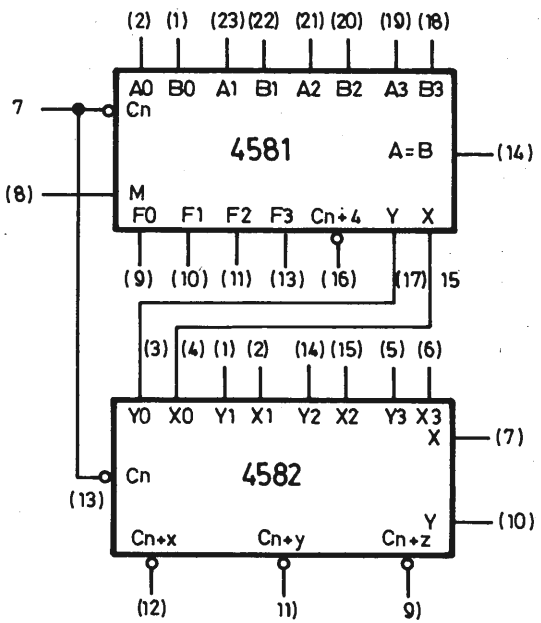


Fig. 9.3 *Collegamento in cascata con il componente 4582; i dati sono in logica negativa.*

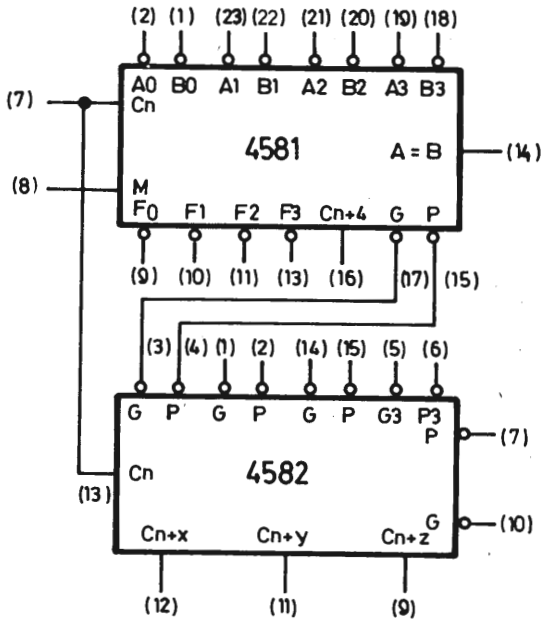


Fig. 9.4 *Collegamento in cascata con il componente 4582; i dati sono in logica positiva.*

III. L'esempio che sviluppiamo di seguito assume il dato A uguale a 0011 e il dato B uguale a 0010, considerando dati e segnali attivi a livello alto. Realizzando tutte le funzioni aritmetiche possibili e mettendo l'entrata C_n a livel-

M = 0 ; DATO A = 0011 ; DATO B = 0010 ; $C_n = 1$						
S3	S2	S1	S0	Funzione	Uscita	Display
0	0	0	0	$F = A$	0011	3
0	0	0	1	$F = A + B$	0101	5
0	0	1	0	$F = A + \overline{B}$	0000	0
0	0	1	1	$F = -1$	1111	bianco
0	1	0	0	$F = A + A\overline{B}$	1010	bianco
0	1	0	1	$F = A + B + A\overline{B}$	0100	4
0	1	1	0	$F = A - B - 1$	0000	0
0	1	1	1	$F = A\overline{B} - 1$	0110	6
1	0	0	0	$F = A + AB$	1000	8
1	0	0	1	$F = A + B$	0101	5
1	0	1	0	$F = A + \overline{B} + AB$	0111	7
1	0	1	1	$F = AB - 1$	0110	6
1	1	0	0	$F = A + A$	0110	6
1	1	0	1	$F = A + B + A$	1000	8
1	1	1	0	$F = A + \overline{B} + A$	0011	3
1	1	1	1	$F = A - 1$	0010	2

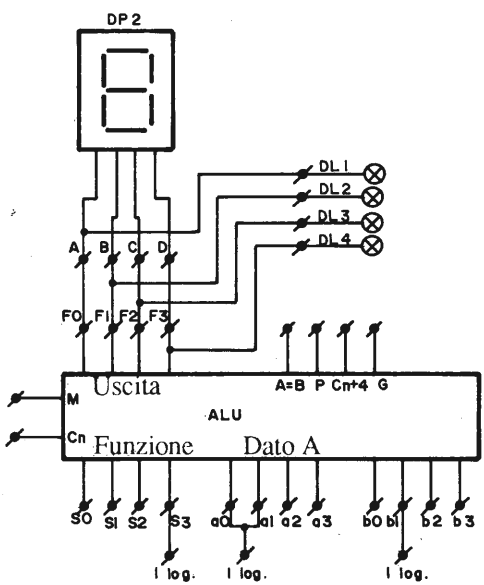


Fig. 9.5 *Schema elettrico dell' esercitazione 1.*

lo alto (senza riporto), otterremo i seguenti risultati:

IV. Adesso svilupperemo lo stesso esempio, ma con l'entrata C_n a livello

M = 0 ; DATO A = 0011 ; DATO B = 0010 ; C_n = 0						
S3	S2	S1	S0	Funzione	Uscita	Display
0	0	0	0	$F = A + 1$	0100	4
0	0	0	1	$F = A + B + 1$	0110	6
0	0	1	0	$F = A + \overline{B} - 1$	1111	bianco
0	0	1	1	$F = 0$	0000	0
0	1	0	0	$F = A + A\overline{B} - 1$	1001	9
0	1	0	1	$F = A + B + A\overline{B} + 1$	0101	5
0	1	1	0	$F = A - B$	0001	1
0	1	1	1	$F = A\overline{B}$	0111	7
1	0	0	0	$F = A + AB + 1$	1001	9
1	0	0	1	$F = A + B + 1$	0110	6
1	0	1	0	$F = A + \overline{B} + AB + 1$	1000	8
1	0	1	1	$F = AB$	0111	7
1	1	0	0	$F = A + A + 1$	0111	7
1	1	0	1	$F = A + B + A + 1$	1001	9
1	1	1	0	$F = A + \overline{B} + A + 1$	0100	4
1	1	1	1	$F = A$	0011	3

basso (con riporto), otterremo i seguenti risultati:

V. Come si può vedere dai risultati riportati in tabella, con questa ALU è possibile eseguire praticamente tutte le operazioni aritmetiche esistenti con due parole di quattro bit. E' conveniente ripetere l'esperienza precedente assegnando due valori diversi ai dati A e B, per comprenderne appieno il funzionamento. Proponiamo di farlo con dato A=0101 e dato B=1010.

CAPITOLO 10

MEMORIA RAM

Una memoria RAM (Random Access Memory) è una memoria a lettura e scrittura, nella quale i tempi necessari per leggere e scrivere un dato sono essenzialmente indipendenti dalla locazione in cui è stato posto. Il dato viene introdotto in una determinata locazione e volendo si può togliere dalla stessa.

Per molti anni le memorie RAM sono state costruite tutte nello stesso modo; con nucleo di ferrite, attualmente le memorie a semiconduttore hanno conquistato il mercato, sono più facili da utilizzare e il loro costo di produzione è decisamente inferiore rispetto a quelle con nucleo di ferrite.

10.1 Memorie RAM

Esistono vari tipi di memorie RAM e possono essere divise in tre categorie fondamentali, in funzione della tecnologia utilizzata per la loro costruzione.

- Bipolare: la tecnologia bipolare è quella usata nei circuiti integrati delle famiglie logiche TTL, ECL, ecc. I circuiti bipolari si chiamano così perchè utilizzano transistor bipolari. Le memorie a semiconduttore bipolari sono normalmente compatibili direttamente con una o più delle famiglie logiche bipo-

lari standard. Sono normalmente le più veloci ma anche le più costose; dissipano molta potenza.

- MOS statiche: sia le memorie MOS statiche che quelle dinamiche utilizzano transistor MOSFET. Tuttavia, nei circuiti statici, i transistor sono utilizzati come amplificatori di corrente continua, invece nei circuiti dinamici si fa un grande uso di capacità per l'immagazzinamento temporaneo.

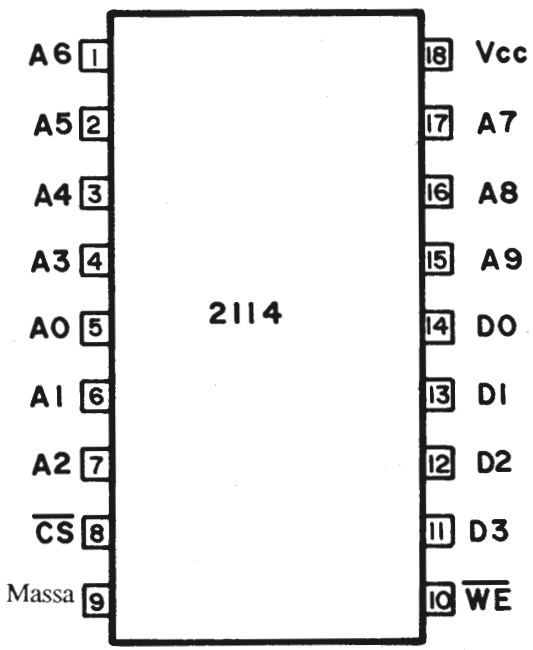


Fig. 10.1 *Piedinatura del circuito integrato 2114.*

In genere, i circuiti MOS statici sono più facili da utilizzare rispetto ai MOS dinamici, ma hanno bisogno di più potenza e hanno un costo superiore. Con le nuove tecnologie MOS, molti circuiti MOS statici possono essere direttamente collegati con circuiti logici delle famiglie TTL o DTL.

I circuiti MOS statici possono essere a canale N o a canale P. Quelli a canale N garantiscono una maggior compatibilità con la famiglia TTL rispetto a

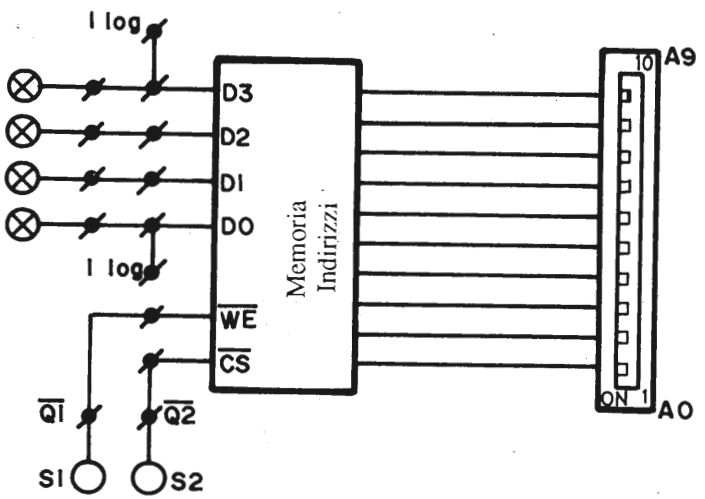


Fig. 10.2 *Schema elettrico dell'esercitazione 1.*

quelli a canale P. Queste memorie sono più lente rispetto alle bipolari, ma costano meno.

- MOS dinamiche: i circuiti MOS dinamici fanno uso dell'immagazzinamento temporaneo dei dati nelle capacità dei circuiti. La carica in queste capacità può sparire nel giro di alcuni millisecondi. Per evitare la perdita dei dati, questa carica deve essere rinfrescata, tale operazione si effettua attraverso la rimemorizzazione dei dati. I circuiti dinamici richiedono frequenze di clock generate esternamente e devono essere normalmente troppo alte per essere generate con circuiti integrati ordinari, sono necessari circuiti integrati speciali o componenti discreti. Da tutto questo si può dedurre che i circuiti dinamici sono più difficili da collegare con le famiglie logiche standard.

Anche se il loro utilizzo è più difficile, questi circuiti hanno anche diversi vantaggi:

- Permettono la memorizzazione di una mole maggiore di dati.

- Le velocità che si possono ottenere con circuiti MOS dinamici sono superiori rispetto a quelli statici, inferiori però rispetto a quelli delle memorie bipolari.

- Sono le memorie che dissipano meno potenza.

Le memorie RAM sono normalmente organizzate in una struttura di N parole composte da M bit ognuna. Poichè normalmente si utilizza il codice binario, l'indirizzo che seleziona una delle N parole è normalmente un valore binario di K bit, N è dunque una potenza di 2 : 2^k.

10.2 Memoria RAM statica 2114

Questo circuito integrato è una memoria RAM statica di 4096 bit organizzati in 1024 parole di quattro bit. E' stata progettata per essere utilizzata con corrente continua stabilizzata (da qui deriva il suo nome di statica), non ha bisogno di clock o rinfresco. Ha la circuiteria necessaria per la decodifica sotto forma di matrice per la localizzazione dei dati.

E' totalmente compatibile con la famiglia TTL.

Una delle sue entrate, il pin 8, identificata da $\overline{CS}$ (chip Select) ci permette di disabilitare il dispositivo.

Viene custodita in un contenitore (in inglese package) di 18 pin.

Caratteristiche

- Tempo massimo di accesso: 450 mS.
- Basso potere di dissipazione: 0,1 mV/bit.
- Non ha bisogno di clock.
- Alimentazione: 5 V ± 5%
- Totalmente compatibile con la famiglia TTL.
- Ingresso e uscita dati comune.
- Immunità al rumore: 400 mV.

ESERCITAZIONE 1. LETTURA E SCRITTURA IN UNA MEMORIA RAM

I. Montare il circuito di Fig. 10.2. Gli ingressi-uscite dati sono fissate attraverso resistenze a massa, cioè a livello basso. Lo sono anche gli indirizzi, cosicchè mettendo tutti gli switch in posizione ON, avremo indirizzato la memoria nella locazione 0..Facendo assumere agli switch tutte le possibili combinazioni potremo indirizzare 1024 parole di quattro bit.

II. In base alla Fig. 10.2 sia il pin 8 ($\overline{CS}$) che il pin 10 ($\overline{WE}$) si trovano a livello alto, il che significa che la memoria non è selezionata ed è in modo di lettura. Proponiamo di iniziare a scrivere all'indirizzo 0 e di inserire il dato 1001 nelle prime locazioni. Per scrivere un dato è necessario premere il pulsante collegato al segnale $\overline{WE}$ e mantenendolo pigiato, premere l'altro per selezionare la memoria. Di seguito si rilasceranno e il dato sarà stato memorizzato nella locazione 0.

Per proseguire le operazioni di scrittura nelle locazioni successive basterà cambiare la posizione degli switch, rammentando che nella posizione OFF si mette il corrispondente segnale a livello alto, in questo modo potremo introdurre altri dati nelle 1024 locazioni di memoria (1K), ognuna di quattro bit.

III. Il passo successivo sarà leggere quello che abbiamo scritto. Per farlo, dobbiamo innanzi tutto scollegare i livelli logici 1 che abbiamo inviato ai pin di ingresso-uscita dati della memoria. Dopo aver eseguito questo passo, basterà tornare a selezionare, attraverso i microinterruttori, gli indirizzi nei quali abbiamo scritto dei dati.

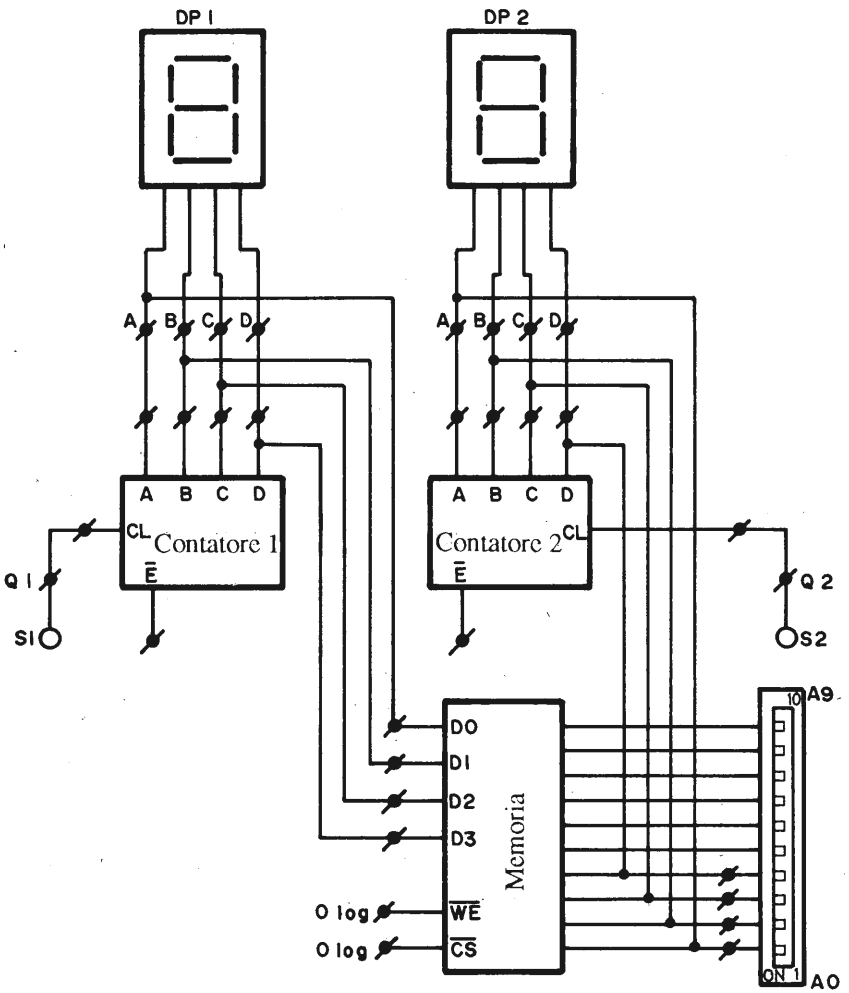


Fig. 10.3 *Schema elettrico dell' esercitazione 2.*

Una volta selezionata la locazione di memoria basterà premere il pulsante collegato al pin $\overline{CS}$, mantenendolo premuto finchè vogliamo visualizzare il dato, verificheremo così che il contenuto è effettivamente quello introdotto precedentemente. Se selezioniamo una locazione qualunque, nella quale, non abbiamo scritto nulla, quando andremo a leggerne il contenuto verificheremo che il risultato è del tutto imprevedibile, potendo assumere un qualsiasi valore da 0 a 16 in binario.

ESERCITAZIONE 2. LETTURA E SCRITTURA IN UNA ZONA DELLA RAM

I. Montare il circuito di Fig. 10.3. Utilizzeremo i due contatori nel seguente modo: il primo è quello che ci fornisce i dati che vogliamo e che visualizziamo sul display corrispondente; il secondo è quello che ci fornisce l'indirizzo della locazione di memoria in cui vogliamo inserire il dato. Mettendo i due segnali, di selezione e scrittura, a livello basso e premendo i pulsanti corrispondenti introdurremo il dato del contatore 1 nella posizione che ci indica il contatore 2, entrambi visualizzati sui display. Tenere presente che quando utilizziamo contatori decimali possiamo lavorare soltanto con valori compresi tra 0 e 9. Per qualsiasi altro dato, inferiore o uguale al numero 15, bisogna introdurlo manualmente come nell'esercitazione precedente.

II. Per verificare i dati precedenti leggiamo la zona di memoria nella quale abbiamo scritto. Per farlo dobbiamo scollegare l'uscita del contatore 1 che va al display 1, in modo tale da visualizzare il contenuto della memoria evitando disturbi da parte del contatore. A questo punto pigeremo ripetutamente il pulsante collegato al contatore 2 in modo tale da visualizzare sul display 2 la locazione di memoria selezionata e sul display 1 il suo contenuto. Tale operazione deve essere compiuta tenendo il segnale $\overline{WE}$ a livello alto e $\overline{CS}$ a livello basso. Per eseguire i cambiamenti necessari per realizzare questa seconda parte, è indispensabile che il segnale $\overline{CS}$ sia a livello alto in modo tale da non alterare il contenuto della memoria.

III. Verifichiamo ora se una memoria RAM è effettivamente volatile. Togliendo l'alimentazione dovremmo perderne il contenuto. Spegnendo e riaccendendo verificheremo che il contenuto di una qualunque locazione è del tutto casuale.

CAPITOLO 11

CODICI ALFANUMERICI E LORO RAPPRESENTAZIONE

Nei codici esposti nei precedenti capitoli, era possibile rappresentare soltanto informazioni numeriche. Ma in molti sistemi digitali, è necessario rappresentare informazione alfabetiche e inoltre alcuni segni speciali, per tanto bisogna utilizzare codici alfanumerici, tra i quali i più utilizzati e importanti sono il codice ASCII e il EBCDIC.

11.1 Codice ASCII

Il codice ASCII (American Standard Code for Information Interchange), che utilizza 7 bit, è riportato nella tabella che segue:

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1									
0	0	0	0		NUL	DLE	SP	0	@	P	'	p
0	0	0	1		SOH	DC1	!	1	A	Q	a	q
0	0	1	0		STX	DC2	"	2	B	R	b	r
0	0	1	1		ETX	DC3		3	C	S	c	s
0	1	0	0		EOT	DC4	$	4	D	T	d	t
0	1	0	1		ENQ	NAK	%	5	E	U	e	u
0	1	1	0		ACK	SYN	&	6	F	V	f	v
0	1	1	1		BEL	ETB	'	7	G	W	g	w
1	0	0	0		BS	CAN	(	8	H	X	h	x
1	0	0	1		HT	EM	)	9	I	Y	i	y
1	0	1	0		LF	SUB	*	:	J	Z	j	z
1	0	1	1		VT	ESC	+	;	K	[	k	{
1	1	0	0		FF	FS	,	<	L	/	l	¦
1	1	0	1		CR	GS	–	=	M	]	m	}
1	1	1	0		SO	RS	.	>	N	^	n	~
1	1	1	1		SI	US	/	?	O	—	o	DEL

NUL - Null
SOH - Start of Heading
STX - Start of Text
ETX - End of Text
EOT - End of Transmission
ENQ - Enquiry
ACK - Acknowledge
BEL - Bell
BS - Back Space
HT - Horizontal Tabulation
LF - Line Feed
VI - Vertical Tabulation
FF - Form Feed
CR - Carriage Return
SO - Shift out
SI - Shift In
SP - Space
DLE - Data link Escape
DC1 - Device Control 1
DC2 - Device Control 2

DC3 - Device Control 3
DC4 - Device Control 4
NAK - Negative Acknowledge
SYN - Synchronous Idle
ETB - End of Transmission Block
CAN - Cancel
EM - End of Medium
SUB - Substitute
ESC - Escape
FS - File Separator
GS - Group Separator
RS - Record Separator
US - Unit Separator
DEL - Delete

11.2 Display

Un display è una rappresentazione visibile dell'informazione. A volte l'informazione si rappresenta nel formato binario, raggruppando i bit di 3 in 3 nel sistema ottale o di 4 in 4 nel sistema esadecimale; ma molte volte è necessaria una rappresentazione decimale, in modo che i risultati possano essere interpretati velocemente.

Per fare un display di questo tipo si possono utilizzare diversi elementi:

- Nixies: sono tubi di gas neon con un solo anodo e 10 catodi diversi. Ogni catodo ha la forma di un digit e sono messi uno dietro l'altro; in questo modo, il catodo che è ionizzato si accende visualizzando il digit corrispondente. Per usare questo elemento di visualizzazione bisogna fornire l'informazione in codice decimale. Hanno lo svantaggio che necessitano di una tensione più alta per il loro funzionamento, inoltre non sono allo stato solido, per questa ragione sono in disuso.

- Diodi luminosi: sono diodi che quando conducono corrente si illuminano. Possono essere alimentati alla stessa tensione del circuito logico. Per fare rappresentazioni numeriche vengono ordinati a forma di 7 segmenti, in caso di rappresentazioni alfanumeriche vengono ordinati a matrice di 7 x 5 punti.

Quando sono a forma di 7 segmenti possono essere costruiti sia ad anodo comune che a catodo comune.

- Cristalli liquidi: quando sono attraversati da corrente riflettono una luce che li illumina. Di solito sono a forma di 7 segmenti e per alimentarli è necessario inviare un'onda quadra di tensione all'ingresso segnato come comune; non può essere corrente continua. Hanno il vantaggio del basso consumo e per questa ragione si utilizzano molto in orologi da polso, calcolatrici alimentate a pile, ecc.

Tutti i Kit presentati nella collana

PROGETTARE CON L'ELETTRONICA

sono disponibili presso:

SYELCO

Via S. Francesco d'Assisi, 20
Via Manzoni, 12
tel. 0321 - 27786 - telefax 0321 - 390254
28100 NOVARA

I kit già disponibili sono elencati nelle pagine seguenti.

I Kit sono comprensivi di tutto il materiale necessario per il montaggio, compresi i circuiti stampati ed esclusi i contenitori, i trasformatori e le minuterie che possono essere richiesti separatamente, così come gli eventuali libri arretrati.

Tutti i prezzi si intendono I.V.A. compresa, franco deposito.

BUONO DI ORDINAZIONE da inviare a:
GRUPPO EDITORIALE JACKSON - Casella postale 10675 - 20100 MILANO

Vi prego di inviarmi:
◊kit completo n._____
◊contenitori relativi al kit n. _____
◊volume PROGETTARE CON L'ELETTRONICA n._____
◊altro _____

Pagherò in contrassegno il prezzo complessivo di L. _____ più le spese postali.

Cognome _____Nome _____
Via _____N. _____
Cap. _____ Città_____Prov._____

ELENCO DEI KIT GIA' DISPONIBILI

PE 001.01	Allarme antifurto (volume 1)	L. 31.000
PE 001.02	Barriera infrarossa (volume 1)	L. 45.000
PE 002.01	Ricevitore via satellite (volume 2)	richiedere
PE 003.01	Volmetro di base (volume 3)	L. 69.500
PE 003.02	Modulo delle scale correnti-tensioni (volume 3)	L. 11.800
PE 003.03	Modulo rettificatore e convertitore f⇒V (volume 3)	L. 23.500
PE 004.01	Alimentatore per preamplificatore (volume 4)	L. 23.000
PE 004.02	Preamplificatore audio (volume 4)	L. 47.500
PE 005.01	Trasmettitore ad infrarossi a (volume 5)	L. 42.000
PE 005.02	Ricevitore ad infrarossi (volume 5)	L. 91.000
PE 006.01	Ricetrasmettitore a tre bande (volume 6)	L.215.000
PE 007.01	Alimentatore (volume 7)	L. 97.000
PE 008.01	Amplificatore da 25 W (volume 8)	L. 38.000
PE 008.02	Amplificatore da 40 W (volume 8)	L. 58.000
PE 008.03	Amplificatore da 100 W (volume 8)	L. 66.000
PE 008.04	Alimentatore per amp. da 25 e 40 W (volume 8)	L. 66.000
PE 008.05	Misuratore di potenza a LED (volume 8)	L. 54.000

Questo libro è la continuazione del primo volume "Elettronica digitale con esperimenti. Dalle porte logiche ai flip-flop", nel quale, a causa del poco spazio a disposizione, non è stato possibile completare quest'ampio argomento. Questo non significa che se non è stato letto il libro precedente, questo non sia possibile capirlo, dato che per la comprensione di questo volume l'unica cosa necessaria è avere già acquisito alcuni principi fondamentali.

Non si vuole neppure esaurire l'argomento, ma approfondire lo studio di alcuni componenti elettronici digitali quali contatori, unità aritmetico logiche e memorie. Con la nuova piastra per esperimenti digitali si completano questi concetti, essendo quest'insieme sperimentale indipendente dal precedente.

Analizzando i due volumi il lettore potrà farsi un'idea chiara sui principi fondamentali dei calcolatori, che costituiscono il mondo dell'elettronica digitale.

ISBN 88-256-0330-4

9 788825 603309

Cod.091P
L.9.500